

Documentation Zenbus

API GTFS-RT

CE DOCUMENT DÉCRIT L'UTILISATION ET LE CONTENU DE L'API GTFS-RT DE ZENBUS



Version	1.0
Date	01/02/2021

Introduction

Deux concepts sont utilisés dans ce document :

- [GTFS](#) : (General Transit Feed Specification) standard le plus utilisé par les services de mobilité d'information voyageur multimodaux. Il est moins riche, mais plus normé que le NeTEx et plus simple à utiliser (plus d'outils compatibles et plus simple de développer ses propres outils)
- [GTFS-RT](#) : GTFS-realtime est une spécification de flux permettant aux agences de transports en commun de fournir aux développeurs d'applications des mises à jour en temps réel sur leur flotte. Il s'agit d'une extension de GTFS (General Transit Feed Specification), un format de données ouvert pour les horaires de transports en commun et les informations géographiques associées

Ouverture

Si vous souhaitez obtenir l'URL du GTFS ou du GTFS-RT de votre réseau, il vous suffit de contacter votre chef de projet ou votre commercial dédié.

Caractéristiques

Fréquence mise à jour	Le flux GTFS-RT est en permanence mis à jour (à la seconde).
Quotas	Aucun. Tout mode de requêtage qui n'est pas déraisonnable (DDoS) est supporté, l'information est générée à la volée, les réponses RT seront potentiellement différentes même à quelques secondes d'intervalle.

Points d'accès

L'URL du GTFS se présente sous la forme suivante :

<code>https://zenbus.net/gtfs/static/download.zip?dataset=<ID_NETWORK></code>
Format : ZIP

L'URL du GTFS-RT se présente sous la forme suivante :

Flux Groupé	<a href="https://zenbus.net/gtfs/rt/poll.proto?dataset=<ID_NETWORK>">https://zenbus.net/gtfs/rt/poll.proto?dataset=<ID_NETWORK>
TripUpdate	<a href="https://zenbus.net/gtfs/rt/poll.proto?file=tu&dataset=<ID_NETWORK>">https://zenbus.net/gtfs/rt/poll.proto?file=tu&dataset=<ID_NETWORK>
VehiclePosition	<a href="https://zenbus.net/gtfs/rt/poll.proto?file=vp&dataset=<ID_NETWORK>">https://zenbus.net/gtfs/rt/poll.proto?file=vp&dataset=<ID_NETWORK>
Format : protocol-buffer	

Protocole d'échange

Le protocole de transport utilisé est le HTTPS.

Services proposés

- TripUpdates
- VehiclePosition

A noter : le service Alert n'est pas encore disponible.

Index des éléments

Zenbus respecte dans sa majorité les spécifications Google. Vous trouverez ci-dessous la liste des champs et des messages que nous implémentons basée sur la [documentation GTFS-RT](#). Nous vous conseillons dès lors de vous référer directement à celle-ci et d'ignorer simplement la partie Alert.

- [FeedMessage](#)
 - [FeedHeader](#)
 - [Incrementality](#)
 - [FeedEntity](#)
 - [TripUpdate](#)
 - [TripDescriptor](#)
 - [ScheduleRelationship](#)
 - [VehicleDescriptor](#)
 - [StopTimeUpdate](#)
 - [StopTimeEvent](#)
 - [ScheduleRelationship](#)
 - [VehiclePosition](#)
 - [TripDescriptor](#)
 - [ScheduleRelationship](#)
 - [VehicleDescriptor](#)
 - [Position](#)
 - [VehicleStopStatus](#)
 - [CongestionLevel](#)
 - [OccupancyStatus](#)

message FeedMessage

Désigne le contenu d'un message de flux. Chaque message du flux consiste en une réponse à une requête HTTP GET appropriée. Un flux en temps réel est toujours défini par rapport à un flux GTFS existant. Tous les identifiants d'entité sont résolus dans le cadre du flux GTFS.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
header	FeedHeader	Oui	One	Métadonnées relatives à ce flux et au message du flux.
entity	FeedEntity	Oui	Many	Contenu du flux. Si des informations en temps réel sont disponibles pour le système de transports en commun, ce champ doit être défini. Si ce champ est vide, les utilisateurs doivent supposer qu'aucune information n'est disponible pour le système.

message FeedHeader

Désigne les métadonnées relatives à un flux incluses dans les messages du flux.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
gtfs_realtime_version	string	Oui	One	Version des spécifications du flux. La version actuelle est 2.0.
incrementality	Incrementality	Oui	One	
timestamp	uint64	Oui	One	Cet horodatage détermine l'heure à laquelle le contenu de ce flux a été créé (à l'heure du serveur). Valeur exprimée à l'heure POSIX (autrement dit, le nombre de secondes depuis le 1er janvier 1970 à 00:00:00 UTC). Pour éviter tout décalage entre les systèmes produisant et consommant des informations en temps

				<p>réel, il est fortement conseillé de définir timestamp en fonction de l'heure du serveur. L'utilisation de serveurs Stratum 3 ou même de strate inférieure est tout à fait acceptée, étant donné que les décalages de quelques secondes au plus sont tolérés.</p>
--	--	--	--	---

enum Incrementality

Détermine si la récupération en cours est incrémentielle.

- **FULL_DATASET** : cette mise à jour de flux écrase toutes les informations en temps réel précédentes pour le flux. Par conséquent, cette mise à jour doit fournir un aperçu complet de toutes les informations en temps réel connues.

Zenbus fournit un flux en FULL_DATASET

- **DIFFERENTIAL** : actuellement, ce mode **n'est pas utilisé** et le comportement **n'est pas spécifié** pour les flux qui utilisent ce mode. La liste de diffusion du flux GTFS-realtime fait l'objet d'une discussion au sujet de la spécification complète du comportement relatif au mode **DIFFERENTIAL**. La documentation sera mise à jour lorsqu'une décision aura été prise.

Valeur
FULL
DIFFERENTIAL

message FeedEntity

Désigne une définition (ou mise à jour) d'une entité dans le flux de transports en commun. Si l'entité n'est pas supprimée, un seul des champs **trip_update**, **vehicle** et **alert** doit être défini.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
id	string	Oui	One	Identifiant de flux unique pour cette entité. Les identifiants sont utilisés uniquement pour fournir un support à l'incrémentalité. Les entités réelles référencées par le flux doivent être spécifiées par des sélecteurs explicites (voir EntitySelector ci-dessous pour en savoir plus).
is_deleted	bool	Non (voir description)	One	Détermine si cette entité doit être supprimée. Ce champ doit être renseigné uniquement pour les flux dont le champ Incrementality est défini sur DIFFERENTIAL . Ce champ ne doit PAS être renseigné pour les flux dont le champ Incrementality est défini sur FULL_ .
trip_update	TripUpdate	Oui	One	Données sur les retards au départ d'un trajet en temps réel. Vous devez renseigner au moins l'un des champs trip_ , vehicle ou alert . Ces champs ne peuvent pas être tous les trois vides.
vehicle	VehiclePosition	Oui	One	Données relatives à la position en temps réel d'un véhicule. Vous devez renseigner au moins l'un des champs trip_ , vehicle ou alert . Ces champs ne peuvent pas être tous les trois vides.

alert	Alert	non	One	Données relatives à l'alerte en temps réel. Vous devez renseigner au moins l'un des champs trip_ , vehicle ou alert . Ces champs ne peuvent pas être tous les trois vides.
-------	-------	-----	-----	---

message TripUpdate

Mise à jour en temps réel de la progression d'un véhicule sur un trajet. Veuillez également consulter la discussion générale sur les [entités des mises à jour de trajet](#).

En fonction de la valeur de **ScheduleRelationship**, le champ **TripUpdate** peut indiquer :

- un trajet qui respecte les horaires indiqués ;
- un trajet qui suit un itinéraire, sans horaire fixe ;
- un trajet qui a été ajouté ou supprimé en fonction des horaires.

Les mises à jour peuvent être appliquées à des arrivées/départs futurs prévus, ou à des événements passés qui se sont déjà produits. Dans la plupart des cas, les informations relatives à des événements passés sont une valeur mesurée, par conséquent, leur valeur d'incertitude recommandée est 0. Il peut néanmoins exister des cas où cette règle ne s'applique pas, il est donc autorisé d'indiquer une valeur d'incertitude différente de 0 pour les événements passés. Si l'incertitude d'une mise à jour n'est pas 0, soit la mise à jour est une prédiction approximative pour un trajet qui ne s'est pas terminé, soit la mesure n'est pas précise ou la mise à jour constituait une prédiction du passé qui n'a pas été validée après la fin de l'événement.

Sachez que la mise à jour peut décrire un trajet déjà effectué. Si c'est le cas, une mise à jour du dernier arrêt du trajet est suffisante. Si l'heure d'arrivée au dernier arrêt est dans le passé, le client en déduira que l'ensemble du trajet est dans le passé (il est néanmoins possible, bien que cela soit sans conséquence, d'effectuer des mises à jour pour les arrêts précédents). Cette option est la plus pertinente pour un trajet qui s'est terminé plus tôt que prévu, même si selon les horaires indiqués, le trajet est toujours en cours à l'heure actuelle. La suppression des mises à jour pour ce trajet peut faire croire au client que le trajet est toujours en cours. Sachez que le fournisseur du flux est autorisé à supprimer définitivement les mises à jour antérieures, mais qu'il n'y est pas obligé. Il s'agit de l'unique cas où cette pratique s'avère concrètement utile.

Nom du	Type	Disponible	Cardinalit	Description
--------	------	------------	------------	-------------

champ		Zenbus	y	
trip	TripDescriptor	Oui	One	Le trajet auquel ce message s'applique. Une limite d'une entité TripUpdate est définie pour chaque instance de trajet réelle. Si aucune entité n'est définie, cela signifie qu'aucune information de prédiction n'est disponible. Cela ne signifie <i>pas</i> que le trajet se déroule selon les horaires prévus.
vehicle	VehicleDescriptor	oui	One	Informations supplémentaires sur le véhicule qui effectue le trajet.
stop_time_update	StopTimeUpdate	oui	Many	Mises à jour de StopTimes (futures, c'est-à-dire les prédictions et dans certains cas, passées, autrement dit celles qui se sont déjà produites). Les mises à jour doivent être triées en fonction du champ stop_ et s'appliquer à tous les arrêts suivants, jusqu'au prochain champ stop_ défini. Au moins un champ stop_ doit être défini pour le trajet, à moins que le champ trip_ ne présente la valeur CANCELED . Si le trajet est annulé, aucun champ stop_ ne doit être défini.
timestamp	uint64	oui	One	Heure à laquelle la progression en temps réel du véhicule a été mesurée. Valeur exprimée à l'heure POSIX (c'est-à-dire le nombre de secondes depuis le 1er janvier 1970 à 00:00:00 UTC).
delay	int32	non	One	Écart actuel entre les horaires prévus et réels du trajet. Le champ delay doit être défini uniquement lorsque la prédiction est

				<p>donnée par rapport à un horaire existant dans le flux GTFS.</p> <p>La valeur du champ delay (exprimée en secondes) peut être positive (le véhicule est en retard) ou négative (le véhicule est en avance). Si delay présente la valeur 0, cela signifie que le véhicule est exactement à l'heure.</p> <p>Les informations concernant les retards contenues dans le champ StopTimeUpdates prévalent sur celles qui sont indiquées au niveau du trajet. Ainsi le retard au niveau d'un trajet est uniquement propagé jusqu'au prochain arrêt présentant une valeur StopTimeUpdate spécifiée dans le champ delay.</p> <p>Il est fortement conseillé aux fournisseurs de flux d'indiquer une valeur TripUpdate. pour spécifier l'heure de la dernière mise à jour de la valeur delay, afin d'évaluer la fraîcheur des données.</p> <p>Attention : Ce champ est encore expérimental et susceptible d'être modifié. À l'avenir, il pourra être officiellement intégré.</p>
--	--	--	--	--

message StopTimeUpdate

Désigne une mise à jour en temps réel des événements d'arrivée et/ou de départ pour un arrêt donné d'un trajet. Veuillez également consulter la discussion générale sur les mises à jour de l'heure des arrêts dans la documentation **TripDescriptor** et entités des mises à jour de trajet.

Les mises à jour peuvent concerner les événements passés et futurs. Le producteur peut ignorer les événements passés (sans y être obligé). La mise à jour concerne un arrêt spécifique, via les champs **stop_sequence** ou **stop_id**. Par conséquent, l'un de ces champs doit nécessairement être défini. Si le même **stop_id** est desservi plus d'une fois dans un trajet, alors,

pour ce trajet, **stop_sequence** doit être fourni dans tous les messages **StopTimeUpdates** pour ce **stop_id** pour ce trajet.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
stop_sequence	uint32	Oui	One	La valeur doit être la même que celle indiquée dans le fichier stop_ dans le flux GTFS correspondant. Vous devez indiquer stop_ ou stop_ dans StopTimeUpdate . Ces deux champs ne peuvent pas être tous les deux vides. Le champ stop_ est obligatoire pour les trajets qui desservent le même stop_ plusieurs fois (lors d'une boucle par exemple), afin d'identifier l'arrêt concerné par cette prédiction.
stop_id	string	oui	One	La valeur doit être la même que celle indiquée dans le fichier stops_ dans le flux GTFS correspondant. Vous devez indiquer stop_ ou stop_ dans StopTimeUpdate . Ces deux champs ne peuvent pas être tous les deux vides.
arrival	StopTimeEvent	oui	One	Si schedule_ est vide ou défini sur SCHEDULED , le champ arrival ou departure doit être indiqué dans StopTimeUpdate . Ces champs ne peuvent pas être tous les deux vides. arrival et departure peuvent être tous les deux vides si schedule_ est défini sur SKIPPED . Si schedule_ est défini sur NO_ , arrival et departure doivent être vides.
departure	StopTimeEvent	oui	One	Si schedule_ est vide ou défini sur SCHEDULED , le champ arrival ou departure doit être indiqué dans

				StopTimeUpdate. Ces champs ne peuvent pas être tous les deux vides. arrival et departure peuvent être tous les deux vides si schedule_ est défini sur SKIPPED . Si schedule_ est défini sur NO_ , arrival et departure doivent être vides.
schedule_	ScheduleRelationship	oui	One	La relation par défaut est SCHEDULED .

enum ScheduleRelationship

Désigne la relation entre ce StopTime et les horaires statiques.

Valeur	Commentaire
SCHEDULED	Le véhicule progresse en respectant les horaires statiques de ses arrêts, bien qu'il ne soit pas forcément conforme aux horaires prévus. Il s'agit du comportement par défaut. Au moins l'un des champs arrival et departure doit être défini.
SKIPPED	L'arrêt est ignoré, c'est-à-dire que le véhicule ne desservira pas cet arrêt. Les champs arrival et departure sont facultatifs.
NO_	Aucune donnée n'est indiquée pour cet arrêt. Cela signifie qu'aucune information en temps réel n'est disponible. Lorsque la valeur NO_ est définie, elle est propagée aux arrêts suivants. Il s'agit donc de la manière recommandée pour identifier l'arrêt à partir duquel vous ne disposez pas d'informations en temps réel. Lorsque la valeur NO_ est définie, aucun des champs arrival ou departure ne doit être renseigné.

message VehiclePosition

Indique les informations relatives à la position d'un véhicule donné en temps réel.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
trip	TripDescriptor	Oui	One	Le trajet que ce véhicule dessert. Ce champ peut être vide ou partiel si le véhicule

				ne peut pas être identifié avec une instance de trajet donnée.
vehicle	VehicleDescriptor	Oui	One	Informations supplémentaires sur le véhicule qui effectue le trajet. Chaque entrée doit présenter un identifiant de véhicule unique.
position	Position	oui	One	Position actuelle de ce véhicule.
current_stop_sequence	uint32	oui	One	L'index de l'ordre des arrêts de l'arrêt actuel. La signification de current_ (c'est-à-dire l'arrêt qu'il désigne) est déterminée par current_ . Si current_ est manquant, la valeur IN_ est supposée.
stop_id	string	non	One	Identifie l'arrêt actuel. La valeur doit être la même que celle indiquée dans le fichier stops. dans le flux GTFS correspondant.
current_status	VehicleStopStatus	oui	One	L'état exact du véhicule au niveau de l'arrêt actuel. Ignoré si current_ est manquant.
timestamp	uint64	oui	One	L'état exact du véhicule au niveau de l'arrêt actuel. Ignoré si current_ est manquant.
congestion_level	CongestionLevel	non	One	
occupancy_status	OccupancyStatus	Oui (dans un futur proche)	One	Taux d'occupation du véhicule. Attention : Ce champ est encore expérimental et susceptible d'être modifié. À l'avenir, il pourra être officiellement intégré.

enum VehicleStopStatus

Valeur	Commentaire
INCOMING_	Le véhicule est sur le point d'arriver à l'arrêt (le symbole du véhicule clignote généralement sur l'affichage de l'arrêt).
STOPPED	Le véhicule est immobilisé à l'arrêt.
NO_	Le véhicule a quitté l'arrêt précédent et poursuit sa route.

enum OccupancyStatus

Désigne le taux d'occupation du véhicule.

Attention : Ce champ est encore **expérimental** et susceptible d'être modifié. À l'avenir, il pourra être officiellement intégré.

Valeur	Commentaire
EMPTY	Le véhicule est considéré comme étant vide selon la plupart des mesures, et compte peu ou pas de passagers. Ceux-ci sont acceptés à bord.
MANY_	Le véhicule présente un grand pourcentage de sièges disponibles. C'est le producteur qui détermine le pourcentage suffisant de sièges inoccupés sur la totalité des sièges disponibles pour appartenir à cette catégorie.
FEW_	Le véhicule présente un faible pourcentage de sièges disponibles. C'est le producteur qui détermine le pourcentage suffisamment faible de sièges inoccupés sur la totalité des sièges disponibles pour appartenir à cette catégorie.
STANDING_	Pour le moment, le véhicule ne peut accueillir que des passagers debout.
CRUSHED_	Pour le moment, le véhicule ne peut accueillir que des passagers debout, et l'espace qui leur est réservé est limité.
FULL	Le véhicule est considéré comme complet selon la plupart des mesures. Toutefois, un petit nombre de passagers est susceptible d'être accepté à bord.
NOT_	Le véhicule ne prend plus de passagers.

message Position

Désigne la position géographique d'un véhicule.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
latitude	float	oui	One	Distance angulaire entre la position et l'équateur, exprimée en degrés, dans le système de coordonnées WGS84.
longitude	float	oui	One	Distance angulaire entre la position et le méridien de référence, exprimée en degrés, dans le système de coordonnées WGS84.
bearing	float	Non (peut-être un jour)	One	Azimut, exprimé en degrés, mesuré dans le sens des aiguilles d'une montre à partir du nord géographique, où 0 indique le nord et 90 indique l'est. Il peut s'agir du relèvement au compas, de la direction vers le prochain arrêt, ou de la position intermédiaire. Cette valeur ne doit pas être déduite de l'ordre des positions précédentes, susceptibles d'être calculées par les clients grâce aux données précédentes.
odometer	double	Non (peut-être un jour)	One	Valeur du compteur kilométrique, exprimée en mètres.
speed	float	non	One	Vitesse instantanée du véhicule, exprimée en mètres par seconde.

message TripDescriptor

Descripteur qui identifie une seule instance de trajet GTFS.

Pour spécifier une instance de trajet unique, il suffit généralement d'utiliser le champ **trip_id** seul. Cependant, dans les situations suivantes, des informations supplémentaires sont nécessaires pour exprimer une instance de trajet unique :

- Si le trajet est défini dans *frequencies.txt*, vous devez obligatoirement ajouter **start_date** et **start_time** en plus de **trip_id**.
- Si le trajet dure plus de 24 heures ou s'il est suffisamment retardé pour entrer en conflit avec un trajet planifié le jour suivant, renseignez à la fois **start_date** et **trip_id**.
- Si vous ne pouvez pas renseigner le champ **trip_id**, les champs **route_id**, **direction_id**, **start_date** et **start_time** sont alors obligatoires.

Dans tous les cas, si **route_id** et **trip_id** sont tous les deux renseignés, la valeur **route_id** doit correspondre à celle du champ **route_id** attribuée au trajet indiqué dans le fichier *trips.txt* du flux GTFS.

Le champ **trip_id** seul, ou combiné à d'autres champs **TripDescriptor**, ne peut pas être utilisé pour identifier plusieurs instances de trajet. Un élément **TripDescriptor** qui correspond à zéro ou à plusieurs instances de trajet (plutôt qu'à une seule instance) est considéré comme erroné. L'entité qui contient l'élément **TripDescriptor** erroné peut être supprimée par les utilisateurs.

Par exemple, lorsqu'un trajet est défini dans le fichier *frequencies.txt* du flux GTFS avec **exact_times=0**, l'élément **TripDescriptor** ne doit jamais spécifier de champ **trip_id** seul. En effet, lorsque vous spécifiez une instance de trajet unique qui démarre à une heure précise de la journée, vous devez également spécifier **start_time**.

Notez que si **trip_id** est inconnu, les identifiants d'ordre des stations figurant dans **TripUpdate** ne sont pas suffisants, et les champs **stop_id** doivent également être définis. En outre, vous devez indiquer des heures d'arrivée et de départ absolues.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
trip_id	string	oui	One	trip_ du flux GTFS auquel ce sélecteur fait référence. Le champ trip_ peut être obligatoire ou non selon le type de trajet : <ul style="list-style-type: none">• Pour les trajets non basés sur la fréquence : Le champ trip_ seul suffit à identifier ces trajets de façon unique.

				<p>Sachez que les trajets non basés sur la fréquence ne sont pas définis dans le fichier <code>frequencies. du flux GTFS</code>.</p> <ul style="list-style-type: none"> • Trajets basés sur la fréquence : les champs trip_, start_ et start_ sont tous obligatoires. Les trajets basés sur la fréquence sont définis dans le fichier <code>frequencies. du flux GTFS</code>. • Pour les trajets basés sur un horaire : le champ trip_ ne peut être omis que si les champs route_, direction_, start_ et start_ combinés permettent d'identifier un trajet de façon unique. Sachez que les trajets basés sur un horaire ne sont pas définis dans le fichier <code>frequencies. du flux GTFS</code>.
route_id	string	oui	One	<p>route_ du flux GTFS auquel ce sélecteur fait référence. Si trip_ est omis, alors les champs route_, direction_, start_ et schedule_ doivent tous être définis pour identifier une instance de trajet.</p>
direction_id	uint32	non	One	<p>Valeur du champ direction_ provenant du fichier <code>trips. du flux GTFS</code>. Elle indique la direction du trajet. Si trip_ est omis, alors direction_ doit être indiqué.</p> <p>Attention : Ce champ est encore expérimental et susceptible d'être modifié. À l'avenir, il pourra être officiellement intégré.</p>
start_time	string	oui	One	<p>Heure de début initialement prévue de cette instance de trajet. Le type de champ Time définit le format de ce champ, par exemple, 11:15:35 ou 25:15:35. start_</p>

				<p>peut être obligatoire ou non selon le type de trajet :</p> <ul style="list-style-type: none"> • Lorsque la valeur du champ trip_id correspond à un trajet non calculé en fonction de la fréquence : le champ start_ doit être omis ou égal à la valeur indiquée pour departure_ dans le fichier stop_ du flux GTFS. • Lorsque la valeur du champ trip_id correspond à un trajet basé sur la fréquence : start_ est obligatoire dans tous les cas, et doit être spécifié pour les mises à jour du trajet et les positions du véhicule. Les trajets basés sur la fréquence sont définis dans le fichier frequencies. du flux GTFS. <ul style="list-style-type: none"> ◦ Si le trajet basé sur la fréquence correspond à un enregistrement GTFS exact_times=1 : le champ start_ doit être un multiple (y compris zéro) de la valeur headway_, et postérieure à la valeur start_ du fichier frequencies. de la période correspondante. ◦ Si le trajet basé sur la fréquence correspond à un enregistrement GTFS exact_times=0 : la valeur du champ start_ peut être arbitraire et est initialement prévue pour correspondre au premier départ du trajet Une fois définie, la valeur start_ de ce trajet exact_ basé sur la fréquence est considérée comme immuable, même si la première heure de départ est modifiée. Les changements d'heure apportés par la suite peuvent être indiqués dans un message StopTimeUpdate. • Si trip_id est omis : start_
--	--	--	--	---

				doit être défini.
start_date	string	oui	One	<p>Date de début de cette instance de trajet, exprimée au format YYYYMMDD. Le champ start_ peut être obligatoire ou non selon le type de trajet :</p> <ul style="list-style-type: none"> • Trajets basés sur un horaire : start_ doit être indiqué. Il permet d'identifier les trajets dont le retard est tel qu'ils entrent en conflit avec un trajet prévu le jour suivant. Prenons l'exemple d'un train qui part à 8:00 et à 20:00 tous les jours. S'il a 12 heures de retard, il y aurait alors deux trajets distincts à la même heure. <p><i>Remarque</i> : Si de tels conflits sont impossibles, ce champ est facultatif pour les trajets basés sur un horaire. Par exemple, pour un service qui propose un départ toutes les heures, si un véhicule qui a une heure de retard n'est plus considéré comme lié à l'horaire initial, alors ce champ est facultatif.</p> <ul style="list-style-type: none"> • Trajets basés sur la fréquence : start_ doit être indiqué. Notez que les trajets basés sur la fréquence sont définis dans le fichier frequencies. du flux GTFS, mais pas ceux basés sur un horaire. • Si trip_id est omis : start_ doit être indiqué.
schedule_relationship	ScheduleRelationship	oui	one	<p>Désigne la relation entre ce trajet et les horaires statiques. Si TripDescriptor est fourni dans un EntitySelector Alert, le champ schedule_ est ignoré par les utilisateurs lorsqu'ils identifient l'instance de trajet correspondante.</p>

enum ScheduleRelationship

Désigne la relation entre ce trajet et les horaires statiques. Si un trajet est effectué conformément aux horaires temporaires, ne figurant pas dans le flux GTFS, il ne doit pas être marqué comme **SCHEDULED**, mais comme **ADDED**.

Valeur	Commentaire
SCHEDULED	Trajet qui est effectué conformément aux horaires GTFS ou qui est suffisamment proche du trajet planifié pour être associé à celui-ci.
ADDED	Trajet supplémentaire qui a été ajouté à des horaires en cours, pour remplacer un véhicule endommagé ou répondre à une hausse soudaine du nombre de passagers, par exemple.
UNSCHEDULED	Trajet qui ne présente pas d'horaires associés. Cette valeur est utilisée pour identifier les trajets définis dans le fichier frequencies. du flux GTFS où exact_. Cette valeur ne doit pas être utilisée pour décrire des trajets non définis dans le fichier frequencies. du flux GTFS, ni ceux définis dans le fichier frequencies. du flux GTFS où exact_.
CANCELED	Trajet qui était prévu dans les horaires, mais qui a été supprimé.

message VehicleDescriptor

Désigne les informations relatives à l'identification du véhicule effectuant le trajet.

Nom du champ	Type	Disponible Zenbus	Cardinalité	Description
id	string	Oui	One	Système interne d'identification du véhicule. Ce champ doit être unique pour chaque véhicule. Il est utilisé pour suivre le véhicule au fil de son évolution dans le système. Cet identifiant ne doit pas être visible pour l'utilisateur final. Utilisez le champ label à cet effet.
label	string	non	One	Libellé visible pour l'utilisateur, c'est-à-dire un élément qui doit être affiché aux passagers afin de leur permettre d'identifier le bon véhicule.
license_plate	string	non	One	Plaque d'immatriculation du véhicule.