

# Mecatran Urbiplan : cahier des charges web-services

octobre 2019

- introduction
- Architecture
  - Sélection du format de sortie
- /ws/server-info
  - Réponse
- /ws/gtfs/feed-info/{feedKey}.
  - Réponse
- /ws/gtfs/agencies/{feedKey}.
  - Paramètres de requête
  - Réponse
- /ws/gtfs/routes/{feedKey}.
  - Paramètres de requête
  - Réponse
- /ws/gtfs/stops/{feedKey}.
  - Paramètres de requête
  - Réponse
- /ws/gtfs/fares/{feedKey}.
  - Réponse
- /ws/realtime/stop/{feedKey}/{stopId}.
  - Paramètres de requête
  - Réponse
- /ws/realtime/vehicles/{feedKey}.
  - Réponse
- /ws/alerts/active/{feedKey}.
  - Paramètres de requête
  - Réponse
- /ws/layer/kml/feed/{feedKey}.
  - Paramètres de requête
  - Réponse

## introduction

Bienvenue dans les spécifications et le guide du développeur des services Web Urbiplan. Urbiplan est une plate-forme offrant une interface RESTful facile à utiliser au client pour accéder aux données de transit statiques et en temps réel. Ce guide explique le fonctionnement de cette API HTTP RESTful (requête, réponse, format JSON ou XML).

**Noter**

Merci de contacter par mail votre point de contact en agence ou [info@mecatran.com](mailto:info@mecatran.com) (<mailto:info@mecatran.com>), pour toute demande technique.

## Architecture

L'API est accessible via de simples requêtes HTTP RESTful telles que :

```
https://app.mecatran.com/utw/ws/gtfs/routes/moontransit?apiKey=xxxxxx
```

Toutes les méthodes nécessitent un `apiKey` paramètre qui est la clé qui aurait dû vous être fournie lors de votre inscription à l'API. Une `apiKey` n'est valable que pour un seul « flux » (généralement une agence de transit). Veuillez vérifier auprès de votre agence les conditions d'utilisation (limitation du débit, transférabilité des clés...) de cette API.

## Sélection du format de sortie

La réponse de l'API est au format JSON ou XML, selon la `Accept`:valeur de l' en-tête HTTP (et la présence du paramètre "callback" dans le cas de JSON vs JSONP). Le paramètre `Accept` : en-tête est *obligatoire* pour toutes les demandes, ne vous fiez pas au format renvoyé par défaut car il peut changer à l'avenir. Nous vous recommandons fortement d'utiliser la version JSON de l'API, car les formats XML et JSONP peuvent être dépréciés à long terme et sont moins utilisés, donc moins bien supportés/testés. Remarque : l'utilisation du format JSONP est également déconseillée, car le serveur gère correctement le mécanisme CORS (préféré).

### **application/json**

Renvoie la réponse au format JSON. La réponse JSON peut être convertie en un objet Javascript assez facilement en code Javascript.

### **application/x-javascript**

Renvoie la réponse au format JSONP. Si vous utilisez ce format, vous devez également ajouter un paramètre "callback" dont le nom est le nom de la fonction que vous souhaitez rappeler. La réponse est un code javascript qui renvoie simplement la fonction de rappel, dont le nom est spécifié à l'aide du paramètre "callback", et avec la charge utile des données comme seul argument.

### **application/xml, text/xml**

Renvoie la réponse au format XML.

**Noter**

Pour tester/débuguer l'API, vous pouvez utiliser l'outil de ligne de commande comme spécifié ci-dessous :

```
curl --header "Accept: application/json" "http://your-url" > debug.json
```

Le proxy utilise la norme JAX-RS pour les services RESTful, implémentée via la bibliothèque EasyREST. L'encodage JSON utilise la bibliothèque Jackson v2. Pour plus d'informations sur Jackson, veuillez consulter [ce lien \(http://jackson.codehaus.org/\)](http://jackson.codehaus.org/).

Le web-service est accessible à l'adresse suivante :

`https://app.mecatran.com/utw/ws/{domain}/{method}/{feedKey}/{objectId}?params=...`

Chaque domaine/méthode de service Web a généralement besoin d'un `feedKey` paramètre de chemin obligatoire qui spécifie le groupe de flux de données que vous souhaitez interroger. Certaines méthodes nécessitent l'ID de l'objet que vous souhaitez interroger. Certaines méthodes acceptent des paramètres de requête facultatifs pour affiner davantage les informations renvoyées.

## /ws/server-info

### Réponse

Divers paramètres de serveur tels que la version. Exemple JSON :

```
{
  "apiVersion" : "1.2.20",
  "version" : "1.4.7"
}
```

**version**

Chaîne de caractères. Version du serveur (major.minor.release)

**apiVersion**

Chaîne de caractères. Version de l'API (major.minor.release) Toutes les versions majeures identiques sont rétrocompatibles, elles ajoutent uniquement de nouveaux champs aux données existantes.

## /ws/gtfs/feed-info/{feedKey}

### Réponse

Un descripteur d'informations de flux statique. Exemple JSON :

```
{
  "id" : "moon-transit",
  "timestamp" : 1426847497598
}
```

**identifiant**

La clé de flux, identique à celle spécifiée dans la requête.

**horodatage**

Un horodatage de la version des données statiques. Si cet horodatage change, cela signifie que les données statiques ont été mises à jour et vous devez mettre à jour toutes les valeurs conservées dans un cache. L'horodatage est également renvoyé dans la requête de départ d'arrêt en temps réel, pour vous aider à minimiser le nombre de demandes au serveur.

## /ws/gtfs/agencies/{feedKey}

## Paramètres de requête

**inclureRoutes**

Valeur booléenne (vrai/faux) spécifiant si vous souhaitez inclure dans la réponse la liste des itinéraires pour chaque agence. La valeur par défaut est « faux ».

## Réponse

Une liste d' transitAgencyobjets. Exemple JSON :

```
[{ "id"      : "moon-rail",
  "lang"    : "eo",
  "name"    : "Moon Rail",
  "phone"   : "(+999) 01 42",
  "timezone": "Moon/Copernicus",
  "url"     : "http://moonrail.com/",
  "routes"  : [{
    "agencyId" : "moon-rail",
    "color"    : "ae6f09",
    "id"       : "01",
    "longName" : "Clavius - Seleucus",
    "shortName": "CS",
    "textColor": "FFFFFF",
    "type"     : 3
  }, {
    ...
  }]
}]
```

**identifiant**

Chaîne de caractères. ID unique interne de l'agence.

**langue**

Code ISO 639-1 à deux lettres de la langue principale de cette agence.

**Nom**

Nom de l'agence.

**téléphone**

Numéro de téléphone de contact de l'agence.

### **fuseau horaire**

Code de fuseau horaire de l'agence. Toutes les dates/heures doivent être calculées par rapport à ce fuseau horaire si aucun autre fuseau horaire n'est spécifié pour un arrêt particulier.

### **URL**

URL du site Web principal de l'agence.

### **itinéraires**

Liste des itinéraires appartenant à ce fuseau horaire. Voir la méthode de liste des routes pour plus d'informations sur la signification des champs d'entité de chaque route.

## **/ws/gtfs/routes/{feedKey}**

## **Paramètres de requête**

### **inclureMotifs**

Valeur booléenne (vrai/faux) spécifiant si vous souhaitez inclure dans la réponse les modèles de trajet et les modèles fusionnés pour chaque itinéraire. Un motif est une liste ordonnée distincte d'arrêts. Les motifs fusionnés sont une vue fusionnée de chaque motif pour une direction, une par direction. La valeur par défaut est « faux ».

### **routeLookAheadDays**

Si défini, filtrez la liste des itinéraires avec uniquement les itinéraires actifs (ayant au moins un trajet en cours) entre maintenant et maintenant + les jours routeLookAheadDays. Si elle n'est pas définie, la liste d'itinéraires n'est pas filtrée.

### **codeQuery, nameQuery**

S'il est présent, restreignez la liste des routes renvoyées à celle contenant la sous-chaîne donnée dans le code (c'est-à-dire le nom court) ou le composant de nom long.

### **latitude, longitude, rayonMètres**

Si ces 3 paramètres sont présents, restreignez la liste des itinéraires renvoyés à celui ayant des arrêts dans le cercle délimité donné, dont le centre est à (latitude, longitude) et ayant des radiusMeters.

## **Réponse**

Une liste d' transitRouteobjets. Exemple JSON :

```
[
  {
    "agencyId": "moon-rail",
    "id": "01",
    "longName": "Clavius - Seleucus",
    "shortName": "CS",
    "color": "AE6F09",
    "textColor": "FFFFFF",
    "type": 3,
    "group": "rail",
    "direction0Name": "Clavius",
    "direction1Name": "Seleucus",
    "fromDate": "2019-07-01T00:00:00Z",
    "toDate": "2020-07-03T00:00:00Z",
    "wheelchairAccessiblePerc": 0,
    "wheelchairUnaccessiblePerc": 0,
    "patterns": [
      {
        "directionId": 0,
        "stops": [ { "stopId": "S34" }, ... { "stopId": "S44" } ]
      }
    ]
  },
  ...
]
```

**identifiant**

Chaîne de caractères. ID unique interne de l'itinéraire.

**ID de l'agence**

Chaîne de caractères. ID interne de l'agence à laquelle appartient cet itinéraire.

**nom court**

Code pour cet itinéraire. Optionnel.

**nom long**

Nom long de la route.

**Couleur**

Couleur de fond de l'itinéraire, en notation standard "HTML" en triplet hexadécimal RVB. Facultatif, blanc par défaut. Exemple : 000000 - noir ; ffffff - blanc; ff0000 - rouge ; 00ff00 - vert ; 0000ff - bleu.

**textCouleur**

Couleur du texte de l'itinéraire. Facultatif, noir par défaut. Même format que la couleur de fond.

**taper**

Type de route (voir les spécifications GTFS pour plus d'informations) :

**0**

Tram, tramway, tramway. Tout système de métro léger ou de niveau de rue dans une région métropolitaine.

**1**

Métro, métro. Tout système ferroviaire souterrain dans une région métropolitaine.

**2**

Rail. Utilisé pour les déplacements interurbains ou longue distance.

**3**

Autobus. Utilisé pour les trajets de bus à courte et longue distance.

**4**

Traversier. Utilisé pour le service de bateau à courte et longue distance.

**5**

Téléphérique. Utilisé pour les téléphériques au niveau de la rue où le câble passe sous la voiture.

**6**

Télécabine, téléphérique suspendu. Généralement utilisé pour les téléphériques où la voiture est suspendue au câble.

**7**

Funiculaire. Tout système de rail conçu pour les pentes raides.

## **motifs**

La liste des modèles pour l'itinéraire. Chaque modèle est une liste ordonnée différente d'arrêts pour différentes directions. Veuillez noter que les modèles qui sont des sous-modèles d'autres ne sont pas fusionnés. Par exemple, si certains trajets ont le modèle ABC et d'autres trajets ont le modèle ABCD, deux modèles sont donnés : ABC et ABCD.

## **patterns/directionId**

La directionId des trajets pour ce modèle. Optionnel. Si les trajets n'ont pas d'ID de direction, leur modèle n'aura pas d'ID de direction. Vous devez gérer le cas où ce champ n'est pas présent.

## **schémas/déplacements**

Nombre total de voyages de ce modèle.

## **patterns/tripDaysCount**

Nombre total de jours pour chaque voyage de ce modèle (la somme du nombre de jours de chaque voyage est applicable). Afin d'obtenir l'importance relative d'un modèle dans un itinéraire, vous pouvez diviser ce nombre par la somme de tripDaysCount pour chaque modèle de l'itinéraire.

## **modèles/identificateurs d'arrêt**

La liste ordonnée des identifiants d'arrêt pour ce modèle.

## **direction0Nom, direction1Nom**

Le nom de la route principale headsign dans les deux directions. Remarque : si cette valeur n'est pas présente dans les données sources, elle est calculée en fonction des panneaux de signalisation du trajet dans chaque sens. La valeur est calculée en

ajoutant, dans l'ordre décroissant d'utilisation, les signes de tête les plus courants (un signe de tête qui est utilisé par plus de 10 % des déplacements), limité aux 3.

### **duDate, auDate**

La première et la dernière date d'activité pour l'itinéraire. Remarque : ces valeurs sont nulles (non présentes) si une route est vide. Le format de date est ISO 8601. Seule la partie date est pertinente, l'heure est toujours minuit UTC quelle que soit l'heure du premier ou du dernier trajet (date logique).

### **fauteuil roulantAccessiblePerc, fauteuil roulantUnaccessiblePerc**

Le pourcentage de voyages-jours avec des voyages accessibles en fauteuil roulant (respectivement, des voyages non accessibles en fauteuil roulant). Remarque : la somme des deux valeurs n'est pas toujours égale à 100 %, car certains trajets peuvent avoir une accessibilité en fauteuil roulant indéfinie, auquel cas elle ne compte dans la somme d'aucune des deux valeurs. Pour considérer qu'un itinéraire est accessible en fauteuil roulant, vous pouvez soit prendre une valeur minimale pour `fauteuil roulantAccessiblePerc` (comme 95% ou 100%), soit prendre une valeur maximale pour `fauteuil roulantUnaccessiblePerc` (comme 0% ou 5%), selon si vous voulez considérer les trajets non définis comme accessibles ou non. Remarque : L'accessibilité en fauteuil roulant est réellement définie trajet par trajet et arrêt par arrêt, et ces valeurs doivent être prises en compte pour être précises (voir les points d'extrémité de l'API arrêts et départs en temps réel).

## **/ws/gtfs/stops/{feedKey}**

## **Paramètres de requête**

### **inclure des arrêts**

Valeur booléenne (vrai/faux) spécifiant si vous souhaitez inclure des arrêts dans la réponse. La valeur par défaut est « vrai ». Un arrêt peut (éventuellement) appartenir à une station mère.

### **inclure des stations**

Valeur booléenne (vrai/faux) spécifiant si vous souhaitez inclure des stations dans la réponse. La valeur par défaut est « vrai ». Une station est un agrégat d'arrêts.

### **inclureRoutes**

Valeur booléenne (true/false) spécifiant si vous souhaitez inclure les détails des routes dans la réponse. La valeur par défaut est « faux ». La liste d'itinéraires d'une station mère est l'union de la liste d'itinéraires de ses enfants (pour la direction, renverra l'union des directions des enfants).

### **routeLookAheadDays**

Si défini, filtrez chaque liste d'itinéraires d'arrêt avec uniquement les itinéraires actifs (ayant au moins un trajet en cours) entre maintenant et maintenant + les jours `routeLookAheadDays`. Si elle n'est pas définie, la liste d'itinéraires n'est pas filtrée.



**codeRequête**

S'il est fourni, ne renvoie que les arrêts dont le code correspond exactement (sensible à la casse et aux accents) à la chaîne codeQuery.

**nomRequête**

S'il est fourni, ne renvoie que les arrêts dont le nom contient la chaîne nameQuery fournie. Ne tenez pas compte des accents ou des majuscules/minuscules. Par exemple, toutes les requêtes "café", "café", "CAFE" ou "CAFÉ" correspondront aux arrêts "Café", "FOO CAFÉ BAR", "cafétéria", etc...

**latitude, longitude, rayonMètres**

Si ces 3 paramètres sont présents, restreignez la liste des arrêts renvoyés à celui du cercle délimité donné, dont le centre est à (latitude, longitude) et ayant des radiusMeters.

## Réponse

Une liste d' transitStopobjets. Exemple JSON :

```
[{
  "id" : "ARIS01",
  "parentId" : "ARIS",
  "latitude" : 47.445178,
  "longitude" : 23.712889,
  "name" : "Aristarchus",
  "type" : 0,
  "wheelchairBoarding" : 0,
  "routeIds" : [ "01" ],
}, {
  "id" : "TYCHO",
  "latitude" : -43.31945,
  "longitude" : 11.36548,
  "name" : "Tycho South",
  "type" : 1,
  "wheelchairBoarding" : 0,
  "routeIds" : [ "01", "02" ],
}, ...]
```

**identifiant**

Chaîne de caractères. ID unique interne de l'arrêt.

**extId**

Chaîne de caractères. ID externe / supplémentaire de l'arrêt, si présent. Optionnel.

**parentId**

Chaîne de caractères. Identifiant interne de la station mère, si cet arrêt appartient à une station. Optionnel.

**latitude Longitude**

Valeur à virgule flottante (double). WGS84 degrés. Coordonnées du point.

**Nom**

Chaîne de caractères. Nom UTF-8 de l'arrêt ou de la station.

### en fauteuil roulant

Entier. Ce champ identifie si les embarquements en fauteuil roulant sont possibles à partir de l'arrêt ou de la gare spécifiés. Le champ peut avoir les valeurs suivantes :

**0**

(Valeur par défaut) indique qu'il n'y a pas d'informations d'accessibilité pour l'arrêt ;

**1**

indique qu'au moins certains véhicules à cet arrêt peuvent être embarqués par un passager en fauteuil roulant ;

**2**

L'embarquement en fauteuil roulant n'est pas possible du tout à cet arrêt.

### taper

Le champ type peut avoir les valeurs suivantes :

**0**

Arrêter. Un endroit où les passagers embarquent ou débarquent d'un véhicule de transport en commun ;

**1**

Gare. Une structure physique ou une zone qui contient un ou plusieurs arrêts.

### routelds

Une liste d'identifiants d'itinéraires passant par cet arrêt. Présent si le `includeRoutesparamètre` est défini.

#### Avertir

Avertissement : Champ obsolète. Veuillez utiliser `routes/id` à la place.

### itinéraires/identifiant

ID de l'itinéraire passant par cet arrêt. Présent si le `includeRoutesparamètre` est défini.

### routes/directionlds

Liste des identifiants de direction pour l'itinéraire passant par cet arrêt. Peut contenir 0, 1 ou les deux, selon l'ensemble des trajets passant par cet arrêt. Présent si le `includeRoutesparamètre` est défini.

## /ws/gtfs/fares/{feedKey}

## Réponse

Une liste d' `transitFareobjets`. Exemple JSON :

```
[ {
  "id" : "STD",
  "agencyId" : "MOONTRANSIT",
  "currencyCode" : "EUR",
  "price" : 1.50,
  "paymentMethod" : 0,
  "transfers" : 0,
  "transfersDurationSec" : 0,
  "rules" : [ {
    "originZoneId" : "TYCHO"
    "destinationZoneId" : "ALBATEGNIUS"
  }, {
    "routeId" : "CLAVIUS"
  } ]
}, ...
```

**identifiant**

Chaîne de caractères. Identifiant unique du tarif.

**ID de l'agence**

Chaîne de caractères. ID de l'agence du tarif. Pas fréquemment utilisé. Optionnel.

**code de devise**

Chaîne de caractères. Code de devise ISO 4217, tel que EUR ou USD.

**le prix**

Flotter. Prix du billet.

**mode de paiement**

Entier. Déterminez si le tarif peut être payé à bord ou avant l'embarquement.

**0**

Le tarif est payé à bord.

**1**

Le tarif doit être payé avant l'embarquement.

**transferts**

Entier. Optionnel. S'il est présent, le nombre maximum de transferts. 0 signifie qu'aucun transfert n'est autorisé. S'il n'est pas présent, un nombre infini de transferts est autorisé.

**transfertsDuréeSec**

Spécifie la durée en secondes avant l'expiration d'un transfert.

**règles**

Optionnel. Une liste de règles pour ce tarif à appliquer. Cela correspond étroitement à la norme tarifaire GTFS. Pour plus d'informations sur le fonctionnement des tarifs, veuillez consulter [ce lien](https://code.google.com/archive/p/googletransitdatafeed/wikis/FareExamples.wiki)

(<https://code.google.com/archive/p/googletransitdatafeed/wikis/FareExamples.wiki>).

**/ws/realtime/stop/{feedKey}/{stopId}**

# Paramètres de requête

## identifiant d'arrêt

L'ID de l'arrêt pour lequel vous souhaitez demander des informations en temps réel. Il peut s'agir de l'ID d'un arrêt (type 0) ou d'une station (type 1). Dans le cas d'une gare, l'agrégation des sous-arrêts (départs, itinéraires et alertes) sera restituée. Le champ stop de la réponse renvoyée correspondra à l'arrêt donné dans la requête.

## vialdArrêt

L'ID de l'arrêt pour restreindre la liste des départs. La liste des départs renvoyée ne contiendra que les trajets passant par cet arrêt (ou par tout sous-arrêt d'une gare si le viaStopId fait référence à une gare). Si cette option est activée, la liste des arrêts desservis n'inclura que les arrêts via, sauf si vous activez l'option includeServedStops auquel cas la liste complète des arrêts desservis sera incluse. Facultatif, ne limitez rien s'il n'est pas défini.

## includeServedStops

Valeur booléenne (vrai/faux) spécifiant si vous souhaitez inclure la liste complète des arrêts desservis pour chaque trajet de départ. Facultatif, la valeur par défaut est "false".

### Avertir

Avertissement! Si l'API fournit des données en temps réel, certaines configurations ne sont pas en mesure de fournir des heures d'arrivée en temps réel pour les ServeStops, auquel cas il n'est pas conseillé d'utiliser cette option. Dans tous les cas, il est préférable de contacter le support technique de Mecatran si vous envisagez d'utiliser cette option.

## Date

Date/heure au format ISO-8601 pour laquelle demander des informations en temps réel pour cet arrêt. Format : yyyy-MM-dd'T'HH:mm:ssZ, comme par exemple : 2015-11-22T12:30:40Z(date/heure définie en UTC), ou 2015-11-22T13:30:40+0100(date/heure définie en CET ou fuseau horaire Europe/Paris). La valeur par défaut est la date/l'heure actuelle. Optionnel.

## inclure des alertes

Valeur booléenne (true/false) spécifiant si vous souhaitez inclure des alertes dans la réponse. La valeur par défaut est « vrai ». Optionnel. Remarque : les alertes ne seront incluses que si elles sont disponibles pour ce flux.

## lookAheadSec

Nombre de secondes pour anticiper les départs. Si ce paramètre est défini, seuls les écarts entre la date/heure fournie et la date/heure + anticipation seront renvoyés dans la réponse. Si non fourni, la valeur par défaut est 86400 (24h). Optionnel.

## Langue préférée

Code de langue à deux lettres ISO 639-1 pour sélectionner la langue préférée des données d'alerte. Veuillez noter que si la langue préférée n'est pas disponible en tant que traduction sur le serveur, le titre et la description de l'alerte seront renvoyés avec une langue par défaut. La valeur par défaut est "en". Optionnel.

## Réponse

Un `transitStopRealTimeInfo` objet. Exemple JSON :

```
{
  "feedTimestamp": 1570209109385,
  "alerts": [
    {
      "activeFrom": "2019-10-07T22:00:00Z",
      "description": "Due to a strike tomorrow, routes CS, TY and XD will not be ru
      ...
    }
  ],
  "departures": [
    {
      "alertIds": [],
      "departureTime": "2019-10-07T11:00:00Z",

```

```

        "directionId": 0,
        "dropoff": 2,
        "exactTime": true,
        "headsign": "Tycho South",
        "pickup": 2,
        "realtime": true,
        "routeId": "CS",
        "theoretical": false,
        "tripId": "VN53",
        "tripIndex": 0,
        "tripIndexMax": 14,
        "wheelchairAccessible": 0
    },
    ...
],
"routes": [
    {
        "agencyId": "moon-rail",
        "id": "01",
        "longName": "Clavius - Seleucus",
        "shortName": "CS",
        "color": "AE6F09",
        "textColor": "FFFFFF",
        "type": 3,
        "group": "rail",
        "direction0Name": "Clavius",
        "direction1Name": "Seleucus",
        "fromDate": "2019-07-01T00:00:00Z",
        "toDate": "2020-07-03T00:00:00Z",
    }
],
"stop": {
    "id": "TC-3",
    "latitude": -43.31,
    "longitude": -11.36,
    "name": "Tycho Central",
    "parentId": "TC",
    "type": 0,
    "wheelchairBoarding": 0
}
}

```

## alertes

Une liste d'alertes, actives pour cet arrêt (actives pour au moins un prochain départ, un itinéraire de passage, ou une alerte générale). Veuillez consulter le service Web de la liste des alertes pour plus d'informations sur les champs d'alerte. Cette liste n'est présente que si l'option de requête "includeAlerts" est définie. Veuillez noter que lorsqu'aucun départ n'est présent dans la réponse pour la date donnée, les alertes actives pour cet arrêt sont présentes, si elles sont actives pour la date sélectionnée.

## horodatage

L'horodatage des données statiques, tel qu'il est renvoyé dans le point de terminaison feed-info. Si cet horodatage change, vous devriez probablement mettre à jour toutes les valeurs mises en cache du côté client (par exemple, une liste d'arrêts ou d'itinéraires conservés dans le cache).

## départs

Une liste de départ de cet arrêt pour un itinéraire donné.

**départs/alertes**

Une liste d'identifiants d'alertes liées à ce départ. Peut être vide si aucune alerte n'est active pour ce départ. L'alerte avec cet ID est présente dans la `alertsliste` de l'`transitStopRealTimeInfo`objet. Cette liste n'est présente que si l'option de requête `"includeAlerts"` est définie.

**départs/heure d'arrivée, départs/heure de départ**

Date/heure UTC ISO 8601 pour l'heure d'arrivée ou de départ, qu'elle soit réelle, prévue ou programmée. Parfois, aucun temps de séjour n'est pris en compte dans le système, donc l'heure d'arrivée et de départ sera la même valeur.

**départs/heure d'arrivée prévue, heure de départ/heure de départ programmée**

Date/heure UTC ISO 8601 pour l'heure d'arrivée ou de départ prévue (théorique). Ce champ n'est présent que lorsque le drapeau `"temps réel"` est défini sur vrai. Si l'indicateur de temps réel n'est pas défini, utilisez les champs d'arrivée et de départ pour l'heure planifiée.

**départs/arrivéeDeviationSec, départs/départDeviationSec**

Nombre de secondes d'écart entre l'heure d'arrivée ou de départ prévue (théorique) et en temps réel (estimée). Ces champs ne sont présents que lorsque le drapeau `"temps réel"` est défini sur vrai et si l'heure de la date de départ ou d'arrivée est présente.

**départs/directionId**

Ce champ permet de classer les départs en deux groupes "logiques" (par exemple entrant/sortant, est/ouest, etc...). 0 ou 1. Facultatif (il faut faire attention à une valeur manquante, qui signifie "direction indéfinie").

**départs/signature**

Headsign du départ.

**départs/routelId**

ID de l'itinéraire pour ce départ. Les données d'itinéraire sont contenues dans la `routesliste`.

**départs/temps réel**

Valeur booléenne. Vrai si ce départ est donné par les données de surveillance en temps réel, faux si les données en temps réel ne sont pas disponibles, auquel cas l'heure de départ prévue est donnée à la place.

**Id de départs/voyage**

Identifiant unique interne du voyage correspondant à ce départ.

**départs/stopId**

L'identifiant de l'escale pour ce départ. Présent uniquement si la demande est faite pour une station ; sinon, l'ID d'un arrêt de départ est toujours l'ID de l'arrêt demandé et n'est donc pas nécessaire.

**départs/prise en charge, départs/dépose**

Le type de prise en charge et de dépose pour ce départ. 0 : régulier (toujours s'arrêter pour le ramassage ou le débarquement, par défaut pour le tram ou le métro). 1 : interdit (en cas de restriction de circulation locale par exemple : le ramassage ou le dépôt le long de l'itinéraire à un moment donné peut être interdit). 2: doit téléphoner à l'agence pour organiser le ramassage/le retour. 3 : Demandez au chauffeur (par défaut pour le bus).

### **départs/desservisEcales**

Une liste des arrêts qui sont desservis par ce voyage. Cette liste est en ordre. La liste commence au premier arrêt après l'arrêt actuel, la partie avant l'arrêt actuel n'est pas incluse. Attention : la liste comprendra des arrêts même si un dépôt est interdit (dépôt de type 1).

### **départs/serviStops/stopId**

L'ID de l'arrêt desservi.

### **départs/desservisEcales/dépose**

Le type de dépose pour cet arrêt (voir les constantes de type de dépose ci-dessus).

### **départs/desservisEcales/arrivéeHeure**

L'heure d'arrivée (horaire ou temps réel, selon le drapeau temps réel) à l'arrêt.

### **départs/arrêts desservis/temps réel**

Le drapeau en temps réel. Vrai si cette heure d'arrivée est en temps réel, faux s'il ne s'agit que d'une heure programmée.

### **départs/serviStops/scheduledArrivalTime**

Si l'indicateur en temps réel est activé (vrai), contient l'heure d'arrivée prévue d'origine. Non présent si le drapeau temps réel n'est pas activé (faux).

### **départs/serviStops/alertIds**

Une liste des identifiants d'alerte applicables pour cet arrêt. Cette liste n'est présente que si l'option de requête "includeAlerts" est définie.

### **départ/fauteuil roulantAccessible**

Entier. Ce champ indique si ce voyage est accessible en fauteuil roulant. Le champ peut avoir les valeurs suivantes :

**0**

(Valeur par défaut) indique qu'il n'y a pas d'informations d'accessibilité ;

**1**

indique que ce trajet peut être utilisé par un fauteuil roulant (si l'arrêt est également accessible) ;

**2**

L'embarquement en fauteuil roulant n'est pas possible pour ce voyage.

### **itinéraires**



Une liste d'objets d'itinéraire d'itinéraires passant par cet arrêt. Veuillez noter que vous pouvez avoir plus de trajets que de départs : tous les trajets sont retournés, même s'il n'y a pas de départs pour ce trajet donné pour la date donnée. Voir l'appel du service Web de la liste des itinéraires pour plus d'informations sur les champs.

### arrêter

Informations sur l'arrêt indiqué. Veuillez consulter le service Web de liste d'arrêt pour plus d'informations sur les champs.

## /ws/realtime/vehicles/{feedKey}

## Réponse

Une liste d' VehiclePositionobjets actuellement actifs. Exemple JSON :

```
{
  "feedTimestamp": 1570209110485,
  "timestamp": 1570450612875,
  "vehiclePositions": [
    {
      "bearing": -143.09293,
      "latitude": -43.234,
      "longitude": -12.34,
      "odometerMeters": 8765.43,
      "stopId": "TC-6",
      "stopSeq": 2,
      "stopStatus": "IN_TRANSIT_TO",
      "timestamp": 1570450607000,
      "tripId": "CS-4567",
      "vehicleId": "Eagle"
    },
    ...
  ]
}
```

### horodatage

L'horodatage des données statiques, tel qu'il est renvoyé dans le point de terminaison feed-info. Si cet horodatage change, vous devriez probablement mettre à jour toutes les valeurs mises en cache du côté client (par exemple, une liste d'arrêts ou d'itinéraires conservés dans le cache).

### horodatage

Horodatage de l'alimentation de la position du véhicule. Cela peut être utilisé comme valeur par défaut pour les horodatages de position de véhicule individuel, s'ils ne sont pas fournis.

### identifiant de véhicule

Obligatoire. Identifiant interne unique du véhicule.

### horodatage

Obligatoire. Horodatage des données en temps réel (horodatage de la dernière heure d'acquisition).

**étiquette, plaque d'immatriculation**

Les deux en option. Étiquette lisible par l'homme et numéro de plaque d'immatriculation du véhicule.

**ID de voyage**

Optionnel. ID du trajet que le véhicule effectue réellement.

**identifiant d'arrêt**

Optionnel. ID de l'arrêt actuel auquel le véhicule est arrêté ou du prochain arrêt visé par le véhicule.

**stopSeq**

Optionnel. Séquence d'arrêt dans le parcours de l'arrêt. Présent uniquement si stopId est présent.

**état d'arrêt**

Optionnel. Etat de l'arrêt en cours/prochain : INCOMING\_AT, STOPPED\_AT ou IN\_TRANSIT\_TO.

**latitude Longitude**

Obligatoire. Coordonnées géographiques WGS84 du véhicule.

**palier**

Optionnel. Relèvement du sens de marche, en degré, du véhicule.

**compteurs kilométriques**

Optionnel. La valeur du compteur kilométrique, en mètres, pour le trajet en cours, du véhicule.

**vitesseMps**

Optionnel. Vitesse du véhicule, en mètres par seconde.

## **/ws/alerts/active/{feedKey}**

## **Paramètres de requête**

**Langue préférée**

Code de langue à deux lettres ISO 639-1 pour sélectionner la langue préférée des données d'alerte renvoyées. Veuillez noter que si la langue préférée n'est pas disponible en tant que traduction sur le serveur, le titre et la description de l'alerte seront renvoyés avec une langue par défaut. La valeur par défaut est "en". Optionnel.

**triArrêts**

S'il est défini sur "true", essaiera de trier les arrêts dans leur ordre "naturel" dans la liste de sortie. N'activez pas cette option si elle n'est pas nécessaire car elle peut potentiellement ralentir l'API. La valeur par défaut est « faux ». Optionnel.

# Réponse

Une liste d' transitAlertsobjets actifs pour la plage de date/heure donnée. Exemple JSON :

```
[
  {
    "activeFrom": "2001-01-08T00:00:00Z",
    "agencies": [],
    "apiPublication": {
      "dateTime": "2019-07-05T10:35:00Z"
    },
    "cause": "CONSTRUCTION",
    "description": "Due to public work (crater erasure), routes CS will not be running",
    "effect": "REDUCED_SERVICE",
    "formattedActiveRange": "From Jan 8, 2001",
    "id": "4356",
    "lang": "en",
    "lastUpdated": "2019-07-08T06:05:32Z",
    "level": 2,
    "publishActiveRange": true,
    "routes": [
      {
        "agencyId": "moon-rail",
        "id": "01",
        "longName": "Clavius - Seleucus",
        "shortName": "CS",
        "color": "AE6F09",
        "textColor": "FFFFFF",
        "type": 3,
        "group": "rail",
        "direction0Name": "Clavius",
        "direction1Name": "Seleucus",
        "fromDate": "2019-07-01T00:00:00Z",
        "toDate": "2020-07-03T00:00:00Z",
      }
    ],
    "splitActiveRange": false,
    "stops": [
      {
        "id": "TC-3",
        "latitude": -43.31,
        "longitude": -11.36,
        "name": "Tycho Central",
        "parentId": "TC",
        "type": 0,
        "wheelchairBoarding": 0
      }
    ],
    "title": "Crater erasure - route CS truncated at Tycho Central"
  },
  ...
]
```

## Alertes de transit

Une liste d'alertes, actives dans la plage donnée.

### identifiant

ID unique interne de l'alerte. Cet identifiant est réutilisé dans les champs d'alertes optionnels pour chaque départ.

**niveau**

Niveau de l'alerte :

**1**

Message d'information ;

**2**

Alerte d'avertissement ;

**3**

Alerte critique.

**langue**

Code de langue à deux lettres ISO 639-1 dans lequel le titre et la description sont écrits. Cette langue peut être différente de la langue préférée donnée en paramètre.

**Titre**

Titre au format texte de l'alerte. Généralement un bref résumé qui peut être affiché comme un en-tête. Notez que ce champ peut être manquant (vide) et vous devez gérer ce cas avec élégance. Optionnel.

**la description**

Description complète au format texte de l'alerte (facultatif). Cette description décrit complètement l'alerte et peut être assez longue.

**actifDe, actifÀ**

UTC ISO 8601 actif de/à la plage de date/heure. Les deux sont optionnels : vous pouvez ainsi obtenir un activeTo sans, à l'inverse, les deux ou aucun. Exemple : 2015-06-23T22:00:00Z (Z est pour le méridien zéro, UTC).

**publierPlageActive**

Valeur booléenne. Si la plage active doit être affichée à l'utilisateur, false dans le cas contraire.

**splitPlageActive**

Valeur booléenne. Si false (valeur par défaut), la plage de dates/heures de/à active est continue. Si vrai, la plage doit être divisée en fonction de la partie horaire, si la plage de dates est supérieure à 2 jours. Par exemple : à partir de = 2 janv. 18 h 00 ; au = 3 janvier, 22h. En mode fractionné (splitActiveRange est vrai), l'alerte est active de 18h à 22h tous les jours du 2 janvier au 3 janvier (ici deux intervalles). En mode non fractionné (splitActiveRange est faux), l'alerte est active du 2 janvier 18 h 00 au 3 janvier 22 h 00 (un intervalle). En mode fractionné, si l'heure de fin est inférieure à l'heure de début, l'intervalle d'heures s'étend sur minuit. Par exemple : à partir de = 2 janv., 22 h 00 ; au = 4 janvier, 4 h 00. En mode split, l'alerte est active de 22h à 4h du matin tous les soirs du 2 au 4 janvier (ici deux nuits). En mode non fractionné, l'alerte est active du 2 janvier à 22 heures au 4 janvier à 4 heures du matin. Pour le calcul de l'heure,

**formatéPlageActive**

Plage active formatée dans la langue définie par le paramètre de requête "preferredLang", telle que (en français) : À partir du 2 nov. 2015.

**agences**

Liste facultative des agences pour lesquelles cette alerte est active. Une alerte active pour une agence est considérée comme une « alerte globale » et doit être affichée comme telle dans une application cliente (par exemple dans un écran principal).

**itinéraires**

Liste facultative des itinéraires pour lesquels cette alerte est active. Si aucun itinéraire n'est sélectionné pour cette alerte, la liste des itinéraires passant par les arrêts sélectionnés est sortie à la place.

**itinéraires/direction**

Champ facultatif, présent si l'itinéraire n'est impacté que pour une direction donnée. Si ce champ n'est pas présent, l'itinéraire est impacté dans les deux sens.

**s'arrête**

Liste facultative des arrêts pour lesquels cette alerte est active.

**causer**

Cause de l'alerte. Choix possibles : UNKNOWN\_CAUSE (1), OTHER\_CAUSE (2), TECHNICAL\_PROBLÈME (3), GRÈVE (4), DÉMONSTRATION (5), ACCIDENT (6), VACANCES (7), MÉTÉO (8), MAINTENANCE (9), CONSTRUCTION ( 10), ACTIVITÉ\_POLICE (11), URGENCE\_MÉDICALE (12).

**effet**

Effet de l'alerte. Choix possibles : NO\_SERVICE (1), REDUCED\_SERVICE (2), SIGNIFICANT\_DELAYS (3), DETOUR (4), ADDITIONAL\_SERVICE (5), MODIFIED\_SERVICE (6), OTHER\_EFFECT (7), UNKNOWN\_EFFECT (8), STOP\_MOVED (9). La valeur "NO\_SERVICE" a une signification particulière et implique qu'il n'y a aucun service sur les éléments impactés (trajets et/ou arrêts).

**apiPublication, canal1Publication**

Une liste d'horodatage UTC à laquelle l'alerte doit commencer à s'afficher pour chaque canal.

**Noter**

Veuillez noter que la sémantique d'avoir routeset/ou d'être stopsdans une alerte est un peu complexe. Voici la signification pour différents cas :

**certaines agences, et tout le reste**

L'alerte est une alerte « globale » pour l'ensemble du réseau d'agences.

**quelques itinéraires mais pas d'arrêts**

L'alerte est applicable pour tous les itinéraires donnés pour l'ensemble de leur gamme.

**quelques arrêts mais aucun itinéraire**

L'alerte est applicable pour tous les arrêts donnés, pour tous les itinéraires de passage.

**les arrêts et les itinéraires**

L'alerte est applicable pour tous les arrêts donnés, uniquement pour les itinéraires donnés. (l'ensemble résultant est l' *intersection* des ensembles d'arrêt et d'itinéraire).

## /ws/layer/kml/feed/{feedKey}

## Paramètres de requête

**inclure des arrêts**

Définissez sur `true` ou `false` si vous souhaitez inclure des arrêts dans la réponse. Seuls les arrêts pour les points de terminaison de saut sont renvoyés. Tous les arrêts sont stockés dans un dossier dédié. Vrai par défaut.

**inclureRoute**

Définissez sur `true` ou `false` si vous souhaitez inclure la géométrie de l'itinéraire dans la réponse. Les segments de géométrie d'itinéraire sont stockés dans un dossier dédié. Vrai par défaut.

**Id de route**

Optionnel. S'il est fourni, filtrez le résultat pour l'ID de route fourni uniquement. Si aucune route ne correspond, renvoyez un KML vide.

**agrégatParStation**

S'il est défini sur `true`, seules les stations (ou les arrêts sans station) sont renvoyés dans le résultat. Veuillez noter que cela peut aboutir au retour d'une géométrie moins précise, car les extrémités du segment géométrique seront accrochées à l'emplacement de la station, qui peut être à un endroit différent de l'emplacement d'arrêt d'origine. Faux par défaut.

**ignoreGéométrie non définie**

S'il est défini sur `true`, ne renvoie pas les segments de géométrie qui ne sont pas définis. Si cette option n'est pas activée, un segment indéfini sera renvoyé sous forme de ligne droite entre les deux fins de course. L'activation de cette option peut entraîner le retour d'une géométrie de routage partielle. La valeur par défaut est `false`.

### **élaguerRedundant**

S'il est défini sur `true`, élaguez une géométrie redondante. Soyez prudent, car l'algorithme d'élagage de la géométrie redondante peut entraîner le retour d'une géométrie imprécise. Une géométrie redondante est un segment de géométrie entre deux arrêts qui ont un chemin alternatif qui n'est pas trop long par rapport à l'original. (Par exemple, un segment [AC] peut être marqué comme redondant si les deux géométries [AB] et [BC] sont définies). La valeur par défaut est `false`.

### **utilisationPruneSeuil**

Une valeur seuil à virgule flottante comprise entre 0 et 1. Tous les segments dont l'utilisation est inférieure à cette valeur ne sont pas renvoyés. Le facteur d'utilisation d'un segment (entre 0 et 1) est le facteur de charge en `trip.days` par rapport à la valeur max de l'itinéraire. Par exemple, si l'itinéraire a une utilisation maximale de 1 000 jours de trajet et un segment de 300, l'utilisation du segment est de 0,3. Cette option peut être utile pour rogner la portion la moins souvent utilisée d'un itinéraire et ne renvoyer que les sections les plus importantes. Soyez prudent, car cette option peut entraîner le retour d'une géométrie non connectée dans certains cas. La valeur par défaut est 0.

### **de à**

Une date au format `YYYY-MM-DD`, par exemple `2016-11-23`. Filtrez les segments qui sont utilisés uniquement en dehors de la plage fournie. Une seule date peut être fournie, par exemple, fournir uniquement `from` filtrera les segments qui ne sont utilisés qu'avant la date fournie. Veuillez noter qu'une seule plage de dates est calculée pour chaque segment. Un segment qui n'est utilisé qu'en juin et septembre par exemple, sera ainsi renvoyé si la plage de départ est juillet-août. Veuillez noter que ces dates sont toutes deux *des dates logiques* (un voyage exécuté à 2 heures du matin est généralement inclus dans le jour logique précédent). Par défaut, aucun filtrage (`null`).

### **inclureUtilisation**

Incluez dans la réponse l'utilisation (facteur de charge) de chaque segment. Veuillez voir ci-dessus pour la définition d'une utilisation de segment.

## **Réponse**

Un fichier KML standard, avec un ou deux dossiers (arrêts, géométrie de l'itinéraire).

---

*Fin du document.*