Mecatran

Urbiplan

web-services specifications

Urbiplan: web-services specifications

Revision History

Revision 1.0	Jun 2013	Initial revision. API version 1.1.
Revision 1.1.1	July 2013	Added "includeRoutes" parameter in /gtfs/stops. Added /server-info. API version 1.1.1
Revision 1.1.2	July 2013	Add JSONP support. Make "Accept:" header param mandatory. API version 1.1.2
Revision 1.1.3	August 2013	Allow station in real-time data. API version 1.1.3
Revision 1.2.0	March 2015	Add timestamp end-point and realtime. API version 1.2.0

Table of Contents

Introduction 1
Architecture 2
Output format selection 2
/ws/server-info 3
Response 3
/ws/gtfs/feed-info/{feedKey} 3
Response 3
/ws/gtfs/agencies/{feedKey} 3
Request parameters 3
Response 4
/ws/gtfs/routes/{feedKey} 5
Response 5
/ws/gtfs/stops/{feedKey} 6
Request parameters 6
Response 6
/ws/realtime/stop/{feedKey}/{stopId} 7
Request parameters 7
Response 8
/ws/alerts/active/{feedKey} 9
Request parameters 9
Response 10

Introduction

Welcome to the Urbiplan web-services specifications and developer guide. Urbiplan is a platform giving an easy-to-use RESTful interface to client to access both static and real-time transit data. This guide explains how this RESTful HTTP API works (request, response, XML/JSON format).

Please email your contact point to the agency or `<info@mecatran.com>` for any technical request.

Architecture

The API is accessible through simple RESTful HTTP requests such as:

```
http://app.mecatran.com/utw/ws/gtfs/stops/moontransit?apiKey=xxxxxx
```

All methods requires an `apiKey` parameter which is the key that should have been provided to you when you registered to the API. An apiKey is valid for one "feed" only (usually a transit agency). Please check with your agency regarding the terms of service (rate limiting, key transferability...) of this API.

Output format selection

The API response can be in XML, JSON or JSONP format, depending on the `Accept:` HTTP header value (and the presence of the "callback" parameter in case of JSON vs JSONP). The Accept: header parameter is *mandatory* for all requests, do not rely on the default returned format as it may change in the future.

Possible `Accept:` values

`application/xml, text/xml`	Return the response in XML format.
`application/json`	Return the response in JSON format. The JSON answer can be converted into a Javascript object quite easily in Javascript code.
`application/x-javascript`	Return the response in JSONP format. If you use this format, you must also add a "callback" parameter which name is the name of the function you want to callback. The answer is a javascript code that simply returns the callback function, whose name is specified using the "callback" parameter, and with the data payload as single argument.

To test/debug the API you can use the `curl` command-line tool as specified below:

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

The proxy uses the JAX-RS standard for RESTful services, implemented through the Jersey library. For more information on the JAX-RS norm please consult this link [http://jax-rs-spec.java.net/] . JSON encoding uses the Jackson library. For more information on Jackson please consult this link [http://jackson.codehaus.org/] .

The web-service is accessible at the following address: `http://app.mecatran.com/utw/ws/{domain}/{method}/{feedKey}/{objectId}?params=...`

Each web-service domain/method usually needs a mandatory `feedKey` path parameter which specifies the data feed bundle you want to query. Some methods require the ID of the object that you want to query. Some methods accept optional query parameters to further refine returned informations.

/ws/server-info

Response

Various server parameters such as version. XML example:

```
<serverInfo>
  <version>1.3.9</version>
  <apiVersion>1.2.0</apiVersion>
</serverInfo>
```

JSON example:

```
{
  "apiVersion" : "1.2.0",
  "version" : "1.3.9"
}
```

Response data fields

version	String. Server version (major.minor.release)
apiVersion	String. API version (major.minor.release) All identical major versions are backward-compatible, they only add new fields to existing data.

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

Response

A static feed information descriptor. JSON example:

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

Response data fields

id	The feed key, identical as the one specified in the request.
timestamp	A timestamp of the static data version. If this timestamp changes, that means that the static data has been updated, and you should update any values kept in a cache. The timestamp is also returned in the real-time stop departure query, to help you minimize the number of requests to the server.

/ws/gtfs/agencies/{feedKey}

Request parameters

Parameters

includeRoutes	Boolean value (true/false) specifying if you want to include in the response the route list for each agency. Default to "false".

Response

A list of `transitAgency` objects. XML example:

```
<transitAgencies>
  <transitAgency>
    <id>moon-rail</id>
    <lang>eo</lang>
    <name>Moon Rail</name>
    <phone>(+999) 01 42</phone>
    <timezone>Moon/Copernicus</timezone>
    <url>http://moonrail.com/</url>
    <routes>
      <agencyId>moon-transit</agencyId>
      <id>01</id>
      <longName>Clavius - Seleucus</longName>
      <shortName>CS</shortName>
      <color>ae6f09</color>
      <textColor>ffffff</textColor>
      <type>3</type>
    </routes>
    ...
  </transitAgency>
</transitAgencies>
```

JSON example:

```
[{ "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
   }, {
       ...
   }]
}]
```

Response data fields

id	String. Internal unique ID of the agency.
lang	ISO 639-1 two-letters code of the main language for this agency.
name	Name of the agency.
phone	Contact phone number of the agency.
timezone	ISO 8601 timezone code of the agency. All date/times should be computed relative to this timezone if no other timezone are specified for a particular stop.

url	Main agency website URL.
routes	List of routes belonging to this timezone. See routes list method for more informations on the meaning of each route entity fields.

/ws/gtfs/routes/{feedKey}

Response

A list of `transitRoute` objects. XML example:

```
<transitRoutes>
  <transitRoute>
    <agencyId>moon-rail</agencyId>
    <id>01</id>
    <longName>Clavius - Seleucus</longName>
    <shortName>CS</shortName>
    <color>ae6f09</color>
    <textColor>ffffff</textColor>
    <type>3</type>
  </transitRoute>
  ...
</transitRoutes>
```

Response data fields

id	String. Internal unique ID of the route.
agencyId	String. Internal ID of the agency which this route is part of.
shortName	Code for this route. Optional.
longName	Route long name.
color	Background color of the route, in hex triplet RGB "HTML" standard notation. Example: 000000 - black; ffffff - white; ff0000 - red; 00ff00 - green; 0000ff - blue.
textColor	Text color of the route. Same format as the background color.
type	Type of route (see GTFS specifications for more informations): 0 Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area. 1 Subway, Metro. Any underground rail system within a metropolitan area. 2 Rail. Used for intercity or long-distance travel. 3 Bus. Used for short- and long-distance bus routes. 4 Ferry. Used for short- and long-distance boat service. 5 Cable car. Used for street-level cable cars where the cable runs beneath the car. 6 Gondola, Suspended cable car. Typically used for aerial cable cars where the car is suspended from the cable. 7 Funicular. Any rail system designed for steep inclines.

/ws/gtfs/stops/{feedKey}

Request parameters

Parameters

includeStops	Boolean value (true/false) specifying if you want to include stops in the response. Default to "true". A stop can (optionally) belong to a parent station.
includeStations	Boolean value (true/false) specifying if you want to include stations in the response. Default to "true". A station is an aggregate of stops.
includeRoutes	Boolean value (true/false) specifying if you want to include route IDs in the response. Default to "false". The route list of a parent station is the union of the route list of its children.

Response

A list of `transitStop` objects. XML example:

```
<transitStops>
  <transitStop>
    <id>ARIS01</id>
    <parentId>ARIS</parentId>
    <latitude>47.445178</latitude>
    <longitude>23.712889</longitude>
    <name>Aristarchus</name>
    <type>0</type>
    <wheelchairBoarding>0</wheelchairBoarding>
    <routeIds>01</routeIds>
  </transitStop>
  <transitStop>
    <id>TYCHO</id>
    <latitude>-43.31945</latitude>
    <longitude>11.36548</longitude>
    <name>Tycho South</name>
    <type>1</type>
    <wheelchairBoarding>0</wheelchairBoarding>
    <routeIds>01</routeIds>
    <routeIds>02</routeIds>
  </transitStop>
  ...
</transitStops>
```

JSON example:

```
[{
    "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" ],
}
```

Response data fields

id	String. Internal unique ID of the stop.
parentId	String. Internal ID of the parent station, if this stop belongs to a station Optional.
latitude, longitude	Floating-point value (double). WGS84 degrees. Coordinates of the point.
name	String. UTF-8 name of the stop or station.
wheelchairBoarding	Integer. This field identifies whether wheelchair boardings are possible from the specified stop or station. The field can have the following values: 0 (Default value) indicates that there is no accessibility information for the stop; 1 indicates that at least some vehicles at this stop can be boarded by a rider in a wheelchair; 2 wheelchair boarding is not possible at all at this stop.
type	The type field can have the following values: 0 Stop. A location where passengers board or disembark from a transit vehicle; 1 Station. A physical structure or area that contains one or more stop.
routeIds	A list of route IDs passing by this stop. Present if the `includeRoutes` parameters is set.

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

Request parameters

Parameters

stopId	The ID of the stop you want to request real-time info. This can be the ID of a stop (type 0), or a station (type 1). In case of a station, the aggregation of sub-stops (departures, routes and alerts) will be returned. The stop field of the returned response will correspond to the stop given in the request.
date	UTC date/time for which to request real-time information for this stop. Format: `yyyy-MM-dd'T'HH:mm:ss'Z'` as for example: `2015-06-21T12:30:40Z` UTC. Default to the current UTC date/time. Optional.
includeAlerts	Boolean value (true/false) specifying if you want to include alerts in the response. Default to "true". Optional. Note: alerts will be included only if they are available for this feed.

lookAheadSec	Number of seconds to look ahead for departures. If this parameter is set, then only departures between the provided date/time and date/time + look ahead will be returned in the answer. If not provided, default to all departures of the current logical day. Optional.
preferredLang	ISO 639-1 two-letters language code to select alert data preferred language. Please note that in case the preferred language is not available as translation on the server, the alert title and description will be returned with a default language. Default to "en". Optional.

Response

A `transitStopRealTimeInfo` object. XML example:

```
<transitStopRealTimeInfo>
  <alerts>
    <activeFrom>2015-06-23T22:00:00Z</activeFrom>
    <activeTo>2015-06-25T20:00:00Z</activeTo>
    <description>
      Due to a strike tomorrow, routes CS, TY and XD will not be running.
    </description>
    <id>9523995</id>
    <lang>en</lang>
    <level>2</level>
    <publishActiveRange>true</publishActiveRange>
    <title>Strike on route CS, TY, and XD.</title>
  </alerts>
  <alerts>
    ...
  </alerts>
  <feedTimestamp>1426857296513</feedTimestamp>
  <departures>
    <alertIds>9523947</alertIds>
    <alertIds>...</alertIds>
    <arrivalTime>2015-06-24T09:04:59Z</arrivalTime>
    <departureTime>2015-06-24T09:04:59Z</departureTime>
    <directionId>0</directionId>
    <headsign>Tycho South</headsign>
    <routeId>CS</routeId>
    <realtime>true</realtime>
    <tripId>3650876</tripId>
  </departures>
  <departures>
    ...
  </departures>
  <routes>
    <agencyId>moon-transit</agencyId>
    <id>01</id>
    <longName>Clavius - Seleucus</longName>
    <shortName>CS</shortName>
    <color>ae6f09</color>
    <textColor>ffffff</textColor>
    <type>3</type>
  </routes>
  <routes>
    ..
  </routes>
  <stop>
    <id>TYCHO</id>
    <latitude>-43.31945</latitude>
    <longitude>11.36548</longitude>
    <name>Tycho South</name>
    <type>1</type>
```

```
    <wheelchairBoarding>0</wheelchairBoarding>
  </stop>
</transitStopRealTimeInfo>
```

Response data fields

alerts	A list of alerts, active for this stop (active for at least a next departure, a route passing by, or a general alert). Please see alerts list web-service for more info on the alert fields.
feedTimestamp	The timestamp of the static data, as returned in the feed-info endpoint. If this timestamp changes, you should probably update any cached values on the client side (for example, a list of stops or routes kept in cache).
departures	A list of departure from this stop for a given route.
departures/alertIds	A list of ID of related alerts for this departure. Can be empty if no alert are linked to this departure. The alert with this ID is present in the `alerts` list of the `transitStopRealTimeInfo` object.
departures/arrivalTime, departures/departureTime	UTC ISO 8601 date/time for the arrival or departure time, either actual, expected or scheduled. Sometimes no dwell time are took into account into the system so arrival and departure time will be the same value.
departures/directionId	This field is used to classify departures in two "logical" groups (for example inbound/outbound, eastward/westward, etc...). 0 or 1. Optional.
departures/headsign	Headsign of the departure.
departures/routeId	ID of the route for this departure. The route data is contained in the `routes` list.
departures/realtime	Boolean value. True if this departure is given by real-time monitoring data, false if real-time data is not available, in which case the scheduled departure time is given instead.
departures/tripId	Internal unique ID of the trip corresponding for this departure.
routes	A list of route objects of routes passing by this stop. Please note that you can have more routes than departures: all routes are returned, even if there is no departures for this given route for the given date. See route list web-service call for more info on fields.
stop	Information on the given stop. Please see stop list web-service for more info on fields.

/ws/alerts/active/{feedKey}

Request parameters

Parameters

lookAheadDays	Number of days to look-ahead for future alerts. All alerts starting before now + this timeframe will be returned. Optional. Default to 7 days. Note:

	There is no limit on the number of days, but if you want to provide an "infinite" time window, please use a bounded value (such as 10000 days) to prevent date computations from overflowing on the server.
preferredLang	ISO 639-1 two-letters language code to select the returned alert data preferred language. Please note that in case the preferred language is not available as translation on the server, the alert title and description will be returned with a default language. Default to "en". Optional.

Response

A list `transitAlerts` objects active for the given date/time range. XML example:

```
<transitAlerts>
  <transitAlert>
    <activeFrom>2015-06-23T22:00:00Z</activeFrom>
    <activeTo>2015-06-25T20:00:00Z</activeTo>
    <description>
      Due to a strike tomorrow, routes CS, TY and XD will not be running.
    </description>
    <id>9523995</id>
    <lang>en</lang>
    <level>2</level>
    <publishActiveRange>true</publishActiveRange>
    <title>Strike on route CS, TY, and XD.</title>
    <routes>
      <agencyId>moon-transit</agencyId>
      <id>01</id>
      <longName>Clavius - Seleucus</longName>
      <shortName>CS</shortName>
      <type>3</type>
    </routes>
    <routes>
      <id>02</id>
      ...
    </routes>
    <routes>
      ...
    </routes>
  </transitAlert>
  <transitAlert>
    ...
    <agencies>
      <id>moon-rail</id>
      <name>Moon Rail</name>
    </agencies>
  </transitAlert>
  <transitAlert>
    ...
    <stops>
      <id>TYCHO</id>
      <latitude>-43.31945</latitude>
      <longitude>11.36548</longitude>
      <name>Tycho South</name>
      <type>1</type>
      <wheelchairBoarding>0</wheelchairBoarding>
      <routeIds>01</routeIds>
      <routeIds>02</routeIds>
    </stops>
  </transitAlert>
  ...
</transitAlerts>
```

Response data fields

transitAlerts	A list of alerts, active in the given range.
id	Internal unique ID of the alert. This ID is reused in the optional alerts fields for each departure.
level	Level of the alert: 1 Informational message; 2 Warning alert; 3 Critical alert.
lang	ISO 639-1 two-letters language code in which title and description are written for. This lang could be different from the preferredLang given as parameter.
title	Title in text format of the alert. Usually a short summary which can be displayed as an header. Note that this field may be missing (empty) and you should handle this case gracefully. Optional.
description	Full description in text format of the alert. This description fully describe the alert and can be quite long.
activeFrom, activeTo	UTC ISO 8601 active from/to date/time range. Both are optional: you can thus get an activeTo w/o from, the other way around, both or none. Example: `2015-06-23T22:00:00Z` (Z is for Zero meridian, UTC).
publishActiveRange	Boolean value. If the active range should be displayed to the user, false otherwise.
agencies	Optional list of agencies this alert is active for. An alert active for an agency is considered as a "global alert" and should be displayed as such in a client application (for example in a main screen).
routes	Optional list of routes this alert is active for.
stops	Optional list of stops this alert is active for.

Please note that the semantics of having `routes` and/or `stops` in an alert is a bit complex. Here is the meaning for various cases:

some agencies, and anything else	The alert is a "global" alert for the whole agency network.
some routes but no stops	The alert is applicable for all given routes for the whole of their range.
some stops but no routes	The alert is applicable for all given stops, for all routes passing by.
both stops and routes	The alert is applicable for all given stops, only for the given routes. (the resulting set is the *intersection* of the stop and route sets).

[End of document.]