MobilityData

GBFS to NeTEx & SIRI v1.0

A canonical mapping produced by MobilityData and DATA4PT

Published on 2022-03-15

Executive Summary

MobilityData and DATA4PT experts worked on this canonical mapping between the General Bikeshare Feed Specification (GBFS) and the Network Timetable Exchange (NeTEx) in order to prepare the release of NeTEx Part 5, which covers the addition of new modes such as bike sharing and ride sharing to NeTex. This detailed documentation aims to support the mobility community to increase the quality of travellers' information.

Experts of both sides also wanted to make their work public and accessible to illustrate their core beliefs: GBFS and NeTEx serve the same goal, are fully interoperable, and both have value in the standard ecology. The same can also be said about NeTEx and the General Transit Feed Specification (GTFS).

By interoperability, we do not mean a dynamic API, but rather the ability to exchange data sets between different systems such that the consuming system can automatically convert the data because precise semantics equivalents have been established for each element and attribute. The two most likely use cases envisaged are; (i) To export NeTEx encoded data stops, operators, mobility services, vehicles, etc and related PT data from a Transmodel based data management system to populate the static content of a GBFS system (or to a trip planner that wishes to show the availability of alternative mode Mobility Services), and (ii) for static data to be exported from a GBFS system to a Transmodel/NeTEx based trip planning system that wants to show the availability of alternative mode Mobility Services.

This canonical mapping aims to be the source of truth for going from one format to the other. It does not presume on the format choice made by data producers, nor it is destined to be an implementation guide. Further documentation on both standards can be found on their respective resources center.

The mapping in this document is made in two steps: a high-level mapping of GBFS to Transmodel elements to compare modeling choices, then mapping tables of GBFS to both NeTEx and SIRI (for real-time information) to support full interoperability between the formats. The latter could be used to build a convertor. Examples of the different encoding of the same data in both GBFS and NeTEx are also given. The mapping is designed to evolve to follow the different versions of each format - hence this being a mapping of NeTEx v1.2.2 which introduces the new modes capabilities. The version of GBFS used for this mapping is v2.2.

The original mapping developed by the NeTEx CEN working group will also be released as an Annex of the publication detailing NeTEx Part 5.

Transmodel, NeTEx and SIRI

Here we provide a brief context for those not familiar with Transmodel based standards.

Transmodel is the European conceptual model for public transport that provides a set of definitions and a data architecture covering all aspects of public transport information. A number of different CEN and national standard data formats are based on Transmodel. NeTEx is an XML format, derived from Transmodel used to exchange actual data sets of different types of static data for different purposes. SIRI is a set of real-time data services, also using XML, for the dynamic exchange of data elements. Thus for efficiency, reference data that does not change very often and that provides a context for the mobility services (such as stops, operators, vehicle types, etc) is exchanged periodically as NeTEx documents Data that needs to be processed in real-time (such as vehicle availability) is exchanged dynamically with the SIRI API; SIRI services general exchange just the real time elements with a context already established by the exchange of static data. For example, static data about stops such as their names, locations, codes, etc can be exchanged with NeTEX. Information on real-time arrivals at   stops is exchanged by the SIRI Stop Monitoring service.

Transmodel / NeTEx mostly use separate elements for separate concepts, making elements uniform and reusable in different applications. For efficiency in real-time applications, SIRI may flatten elements into specific views for a specific service - as is often the case with GBFS and GTFS. For example GBFS uses a quite different representation of a stop, agency/operator etc from GTFS.

Transmodel is systematically modularised into packages (the same modularisation is used for the corresponding NeTEx elements), so that an implementation only need use the relevant components.

Whereas GBFS nests data as different structures in a JSON format using a wrapper element providing shared metadata, NeTEX uses XML, organising data elements within a container element called a VERSION FRAME, also providing shared metadata. An XML document may be validated automatically with a validator which will check for data types, syntax, and referential integrity within the document.

Transmodel documentation uses certain conventions to aid the reader. Concepts are given in upper case, for example MOBILITY SERVICE. The corresponding data element in NeTEx or SIRI is shown in ‘CamelCase’, e.g. ***MobilityService***. The Transmodel model is described using the ISO Unified Modelling language (UML) - as class diagrams. In Transmodel UML diagrams colours are used consistently to distinguish the functional model to which an element belongs.

As with other Transmodel mappings to other formats, here we describe just those elements relevant for GBFS, but there are of course many other functional areas covered by Transmodel /NeTEx.

A.1 General Bikeshare Feed Specification (GBFS)

The General Bikeshare Feed Specification, known as GBFS, provides an open data format for exchanging information about cycle sharing and hiring schemes.

GBFS includes reference data on the operators, bike stations, services, etc. available in a locality equivalent to the Transmodel / NeTEx concepts of OPERATOR, STOP PLACE, MOBILITY SERVICE.

GBFS also includes “price plan” data which can be mapped to the Transmodel / NeTEx concepts of FARE PRODUCT and FARE PRICE. A NeTEx implementation based on the Transmodel fare model would normally also describe more precisely the spatial and temporal scope of the access rights granted by the purchased product, using a TARIFF, FARE STRUCTURE ELEMENT, GEOGRAPHICAL INTERVAL TIME INTERVAL, etc. NeTEx can also describe the FARE PRODUCT in detail, for example who is eligible to buy it (USER PROFILE), where and how it may be purchased, (SALES OFFER PACKAGE). NeTEx describes prices, which may be dynamic or preset using a separate FARE PRICE element. The revised GBFS 2.2 specification allows for stepped price bands equivalent to the Transmodel / NeTEx concepts of GEOGRAPHICAL INTERVAL, GEOGRAPHICAL INTERVAL PRICE and TIME INTERVAL, TIME INTERVAL PRICE.

GBFS also includes real-time availability data on cycles – real-time availability is not covered by NeTEx but is described in the Transmodel model and implemented by the SIRI Facility Monitoring service. Historic data on the real time availability status for stations and vehicles can be recorded in NeTEx.

This comparison of GBFS with NeTEx and SIRI is based on the v2.2 of the GBFS specification which was released in April 2021 and introduced a number of new features including “vehicle types”, “geofencing zones” and “Pricing intervals”. https://gbfs.mobilitydata.org/specification

A.1.1 GBFS data model

A.1.1.1 GBFS "physical model" - overview

The following figure gives an overview of the data elements of the GBFS model, inferred from their JSON structures and coloured with the hues used for equivalent Transmodel concepts so that a ready comparison can be made. ..

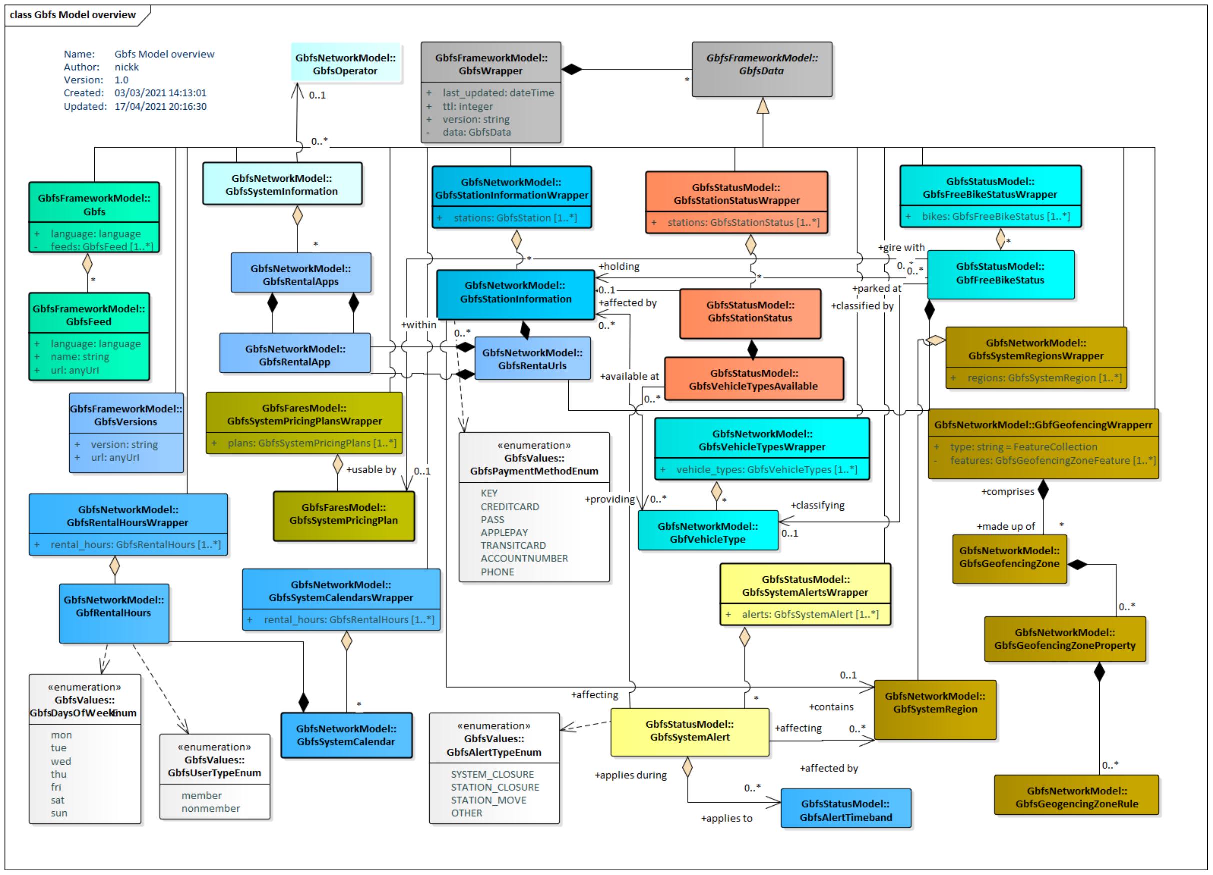

class Gbfs Model overview
Name: Gbfs Model overview
Author: nickk
Version: 1.0
Created: 03/03/2021 14:13:01
Updated: 17/04/2021 20:16:30
GbfsNetworkModel::GbfsOperator
GbfsFrameworkModel::GbfsWrapper
+ last_updated: dateTime
+ ttl: integer
+ version: string
+ data: GbfsData
GbfsFrameworkModel::GbfsData
GbfsFrameworkModel::Gbfs
+ language: language
- feeds: GbfsFeed [1..*]
GbfsFrameworkModel::GbfsFeed
+ language: language
+ name: string
+ url: anyUrl
GbfsNetworkModel::GbfsSystemInformation
GbfsNetworkModel::GbfsStationInformationWrapper
+ stations: GbfsStation [1..*]
GbfsStatusModel::GbfsStationStatusWrapper
+ stations: GbfsStationStatus [1..*]
GbfsStatusModel::GbfsFreeBikeStatusWrapper
+ bikes: GbfsFreeBikeStatus [1..*]
GbfsNetworkModel::GbfsRentalApps
GbfsNetworkModel::GbfsStationInformation
GbfsStatusModel::GbfsStationStatus
GbfsStatusModel::GbfFreeBikeStatus
GbfsNetworkModel::GbfsRentalApp
GbfsNetworkModel::GbfsRentaUrls
GbfsStatusModel::GbfsVehicleTypesAvailable
GbfsNetworkModel::GbfsSystemRegionsWrapper
+ regions: GbfsSystemRegion [1..*]
GbfsFrameworkModel::GbfsVersions
+ version: string
+ url: anyUrl
GbfsFaresModel::GbfsSystemPricingPlansWrapper
+ plans: GbfsSystemPricingPlans [1..*]
«enumeration» GbfsValues::GbfsPaymentMethodEnum
KEY
CREDITCARD
PASS
APPLEPAY
TRANSITCARD
ACCOUNTNUMBER
PHONE
GbfsNetworkModel::GbfsVehicleTypesWrapper
+ vehicle_types: GbfsVehicleTypes [1..*]
GbfsNetworkModel::GbfGeofencingWrapperr
+ type: string = FeatureCollection
- features: GbfsGeofencingZoneFeature [1..*]
GbfsNetworkModel::GbfsRentalHoursWrapper
+ rental_hours: GbfsRentalHours [1..*]
GbfsFaresModel::GbfsSystemPricingPlan
GbfsNetworkModel::GbfVehicleType
GbfsNetworkModel::GbfsGeofencingZone
GbfsStatusModel::GbfsSystemAlertsWrapper
+ alerts: GbfsSystemAlert [1..*]
GbfsNetworkModel::GbfRentalHours
GbfsNetworkModel::GbfsSystemCalendarsWrapper
+ rental_hours: GbfsRentalHours [1..*]
GbfsNetworkModel::GbfsGeofencingZoneProperty
«enumeration» GbfsValues::GbfsDaysOfWeekEnum
mon
tue
wed
thu
fri
sat
sun
«enumeration» GbfsValues::GbfsUserTypeEnum
member
nonmember
GbfsNetworkModel::GbfsSystemCalendar
«enumeration» GbfsValues::GbfsAlertTypeEnum
SYSTEM_CLOSURE
STATION_CLOSURE
STATION_MOVE
OTHER
GbfsStatusModel::GbfsSystemAlert
GbfsNetworkModel::GbfsSystemRegion
GbfsStatusModel::GbfsAlertTimeband
GbfsNetworkModel::GbfsGeogencingZoneRule
+holding
+affected by
+within
+available at
+gire with
+parked at
+classified by
+usable by
+providing
+classifying
+comprises
+made up of
+affecting
+contains
+affected by
+applies during
+applies to

Figure 1 — **GBFS Model (UML)**

The GBFS model includes JSON wrapper elements that are syntactic artefacts and can be ignored in a semantic analysis. (The different GBFS functional elements can all be regarded as specialisations of a common Ur-element ***GbfsData*** that is given metadata by means of a ***GbfsWrapper*).** The following three diagrams show the functional data elements and relationships of the GBFS model (ignoring JSON wrappers). For readability, these are broken down into static, pricing and real-time functional areas, with elements similarly coloured with the hues used for the equivalent Transmodel concepts.

A.1.1.2 GBFS Static network elements

The GBFS data structures have elements to represent static network concepts such as stations and topographic areas. There are also some metadata elements giving information about the feed.

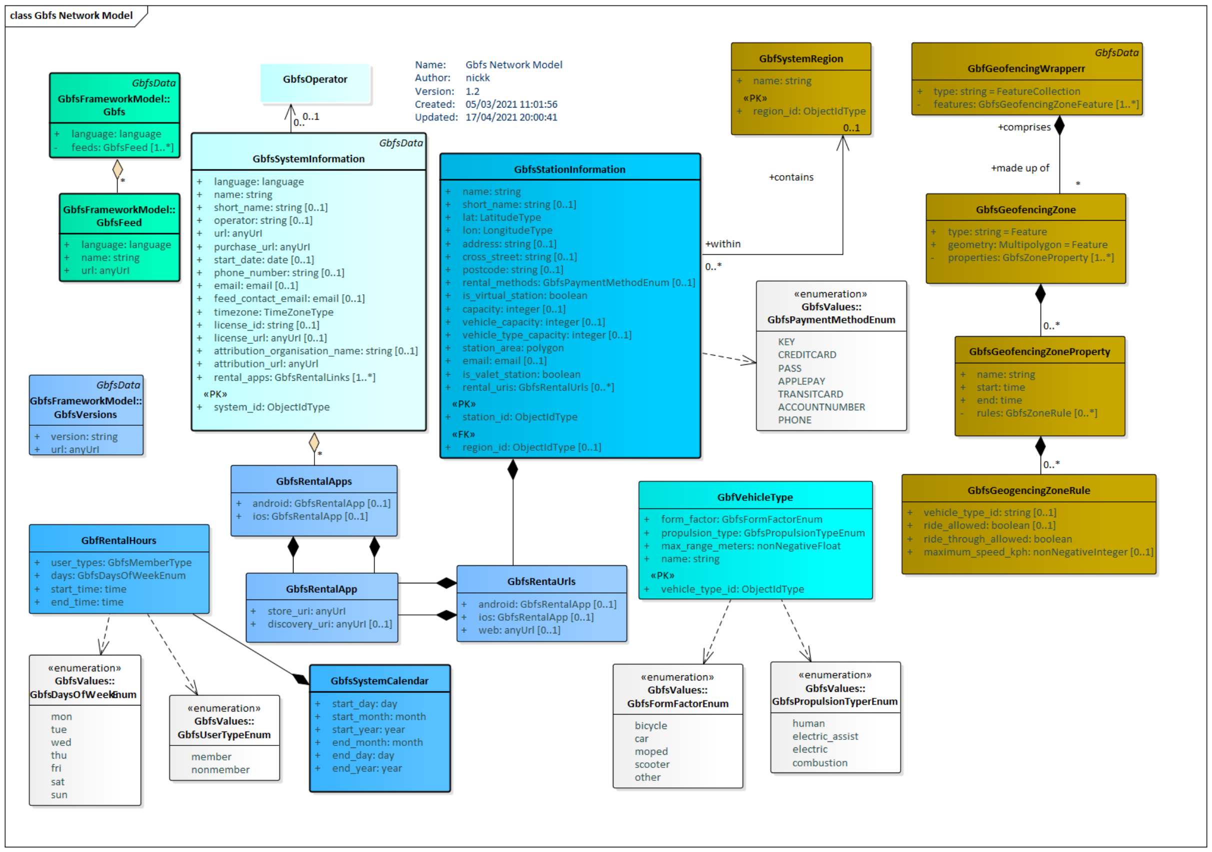

class Gbfs Network Model
Name: Gbfs Network Model
Author: nickk
Version: 1.2
Created: 05/03/2021 11:01:56
Updated: 17/04/2021 20:00:41
GbfsData
GbfsFrameworkModel::Gbfs
+ language: language
- feeds: GbfsFeed [1..*]
GbfsFrameworkModel::GbfsFeed
+ language: language
+ name: string
+ url: anyUrl
GbfsData
GbfsFrameworkModel::GbfsVersions
+ version: string
+ url: anyUrl
GbfsOperator
0..1
GbfsData
GbfsSystemInformation
+ language: language
+ name: string
+ short_name: string [0..1]
+ operator: string [0..1]
+ url: anyUrl
+ purchase_url: anyUrl
+ start_date: date [0..1]
+ phone_number: string [0..1]
+ email: email [0..1]
+ feed_contact_email: email [0..1]
+ timezone: TimeZoneType
+ license_id: string [0..1]
+ license_url: anyUrl [0..1]
+ attribution_organisation_name: string [0..1]
+ attribution_url: anyUrl
+ rental_apps: GbfsRentalLinks [1..*]
«PK»
+ system_id: ObjectIdType
GbfsStationInformation
+ name: string
+ short_name: string [0..1]
+ lat: LatitudeType
+ lon: LongitudeType
+ address: string [0..1]
+ cross_street: string [0..1]
+ postcode: string [0..1]
+ rental_methods: GbfsPaymentMethodEnum [0..1]
+ is_virtual_station: boolean
+ capacity: integer [0..1]
+ vehicle_capacity: integer [0..1]
+ vehicle_type_capacity: integer [0..1]
+ station_area: polygon
+ email: email [0..1]
+ is_valet_station: boolean
+ rental_uris: GbfsRentalUrls [0..*]
«PK»
+ station_id: ObjectIdType
«FK»
+ region_id: ObjectIdType [0..1]
GbfSystemRegion
+ name: string
«PK»
+ region_id: ObjectIdType
0..1
+contains
+within
0..*
GbfsData
GbfGeofencingWrapperr
+ type: string = FeatureCollection
- features: GbfsGeofencingZoneFeature [1..*]
+comprises
+made up of
GbfsGeofencingZone
+ type: string = Feature
+ geometry: Multipolygon = Feature
+ properties: GbfsZoneProperty [1..*]
0..*
GbfsGeofencingZoneProperty
+ name: string
+ start: time
+ end: time
- rules: GbfsZoneRule [0..*]
0..*
GbfsGeogencingZoneRule
+ vehicle_type_id: string [0..1]
+ ride_allowed: boolean [0..1]
+ ride_through_allowed: boolean
+ maximum_speed_kph: nonNegativeInteger [0..1]
«enumeration» GbfsValues::GbfsPaymentMethodEnum
KEY
CREDITCARD
PASS
APPLEPAY
TRANSITCARD
ACCOUNTNUMBER
PHONE
GbfsRentalApps
+ android: GbfsRentalApp [0..1]
+ ios: GbfsRentalApp [0..1]
GbfRentalHours
+ user_types: GbfsMemberType
+ days: GbfsDaysOfWeekEnum
+ start_time: time
+ end_time: time
GbfsRentalApp
+ store_uri: anyUri
+ discovery_uri: anyUrl [0..1]
GbfsRentaUrls
+ android: GbfsRentalApp [0..1]
+ ios: GbfsRentalApp [0..1]
+ web: anyUrl [0..1]
GbfVehicleType
+ form_factor: GbfsFormFactorEnum
+ propulsion_type: GbfsPropulsionTypeEnum
+ max_range_meters: nonNegativeFloat
+ name: string
«PK»
+ vehicle_type_id: ObjectIdType
«enumeration» GbfsValues::GbfsDaysOfWeekEnum
mon
tue
wed
thu
fri
sat
sun
«enumeration» GbfsValues::GbfsUserTypeEnum
member
nonmember
GbfsSystemCalendar
+ start_day: day
+ start_month: month
+ start_year: year
+ end_month: month
+ end_day: day
+ end_year: year
«enumeration» GbfsValues::GbfsFormFactorEnum
bicycle
car
moped
scooter
other
«enumeration» GbfsValues::GbfsPropulsionTyperEnum
human
electric_assist
electric
combustion

Figure 2 — **GBFS Network Model (UML)**

A.1.1.3 GBFS Pricing Plan elements`

The GBFS data structures have elements to represent the prices of vehicle hire. These conflate the Transmodel concepts of FARE PRODUCT, FARE PRICE, TARIFF, access right , user type, etc., etc into simplified views.

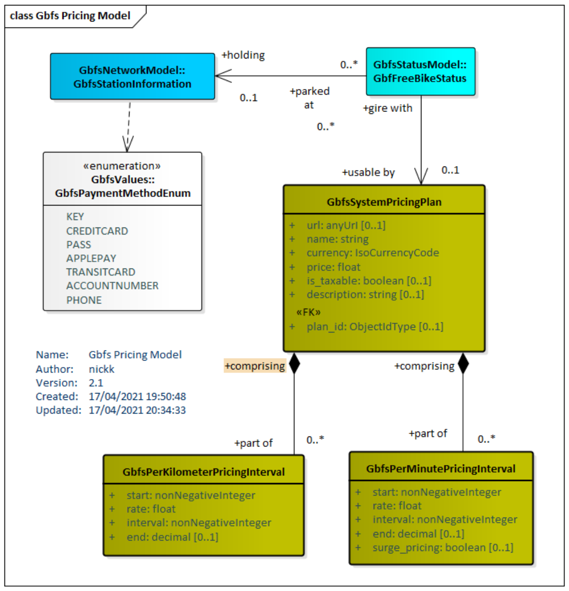

class Gbfs Pricing Model
GbfsNetworkModel:: GbfsStationInformation
GbfsStatusModel:: GbfFreeBikeStatus
+holding
0..*
0..1
+parked at
0..*
+gire with
+usable by
0..1
«enumeration» GbfsValues:: GbfsPaymentMethodEnum
KEY
CREDITCARD
PASS
APPLEPAY
TRANSITCARD
ACCOUNTNUMBER
PHONE
GbfsSystemPricingPlan
+ url: anyUrl [0..1]
+ name: string
+ currency: IsoCurrencyCode
+ price: float
+ is_taxable: boolean [0..1]
+ description: string [0..1]
«FK»
+ plan_id: ObjectIdType [0..1]
Name: Gbfs Pricing Model
Author: nickk
Version: 2.1
Created: 17/04/2021 19:50:48
Updated: 17/04/2021 20:34:33
+comprising
+comprising
+part of
0..*
+part of
0..*
GbfsPerKilometerPricingInterval
+ start: nonNegativeInteger
+ rate: float
+ interval: nonNegativeInteger
+ end: decimal [0..1]
GbfsPerMinutePricingInterval
+ start: nonNegativeInteger
+ rate: float
+ interval: nonNegativeInteger
+ end: decimal [0..1]
+ surge_pricing: boolean [0..1]

Figure 3 — **GBFS Pricing Model (UML)**

A.1.1.4 GBFS Real-time status elements

The GBFS data structures have elements to represent the real-time status of hire stations and vehicles. Equivalent real-time data can be exchanged by the SIRI Facility Monitoring service. Note, however that for historical analysis, real-time values may be recorded as NeTEx LOG ENTRY structures (these will typically be exchanged asynchronously after the event).

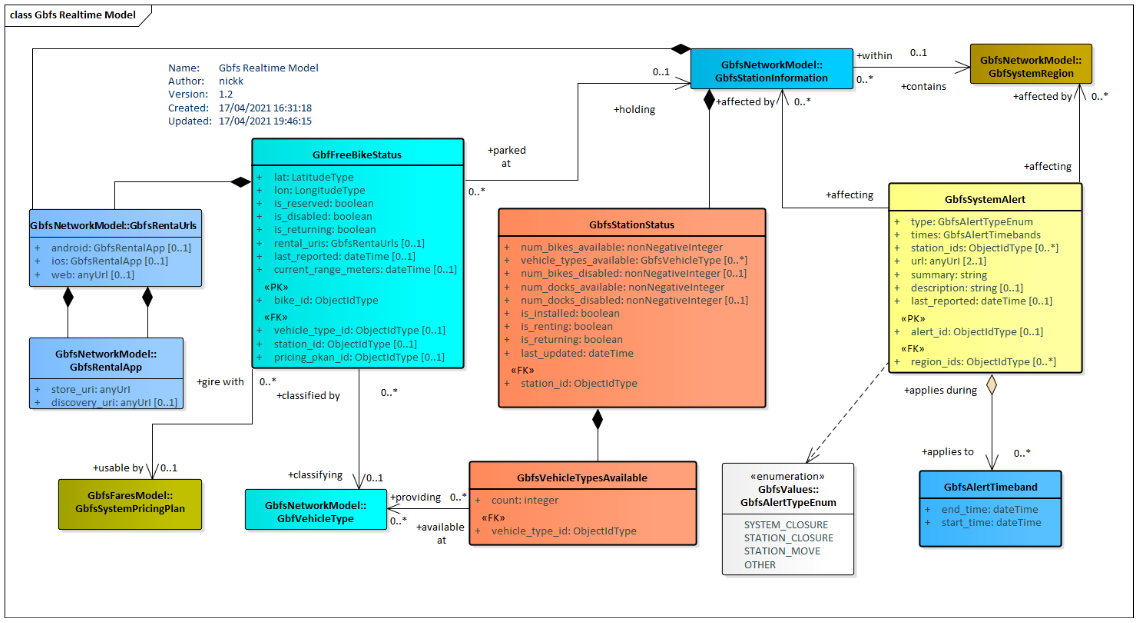

class Gbfs Realtime Model
Name: Gbfs Realtime Model
Author: nickk
Version: 1.2
Created: 17/04/2021 16:31:18
Updated: 17/04/2021 19:46:15
GbfsNetworkModel:: GbfsStationInformation
+within
0..1
0..*
+contains
GbfsNetworkModel:: GbfSystemRegion
+affected by
0..*
0..1
+holding
+affected by
0..*
GbfFreeBikeStatus
+ lat: LatitudeType
+ lon: LongitudeType
+ is_reserved: boolean
+ is_disabled: boolean
+ is_returning: boolean
+ rental_uris: GbfsRentaUrls [0..1]
+ last_reported: dateTime [0..1]
+ current_range_meters: dateTime [0..1]
«PK»
+ bike_id: ObjectIdType
«FK»
+ vehicle_type_id: ObjectIdType [0..1]
+ station_id: ObjectIdType [0..1]
+ pricing_pkan_id: ObjectIdType [0..1]
+parked at
0..*
+affecting
+affecting
GbfsNetworkModel::GbfsRentaUrls
+ android: GbfsRentalApp [0..1]
+ ios: GbfsRentalApp [0..1]
+ web: anyUrl [0..1]
GbfsStationStatus
+ num_bikes_available: nonNegativeInteger
+ vehicle_types_available: GbfsVehicleType [0..*]
+ num_bikes_disabled: nonNegativeInteger [0..1]
+ num_docks_available: nonNegativeInteger
+ num_docks_disabled: nonNegativeInteger [0..1]
+ is_installed: boolean
+ is_renting: boolean
+ is_returning: boolean
+ last_updated: dateTime
«FK»
+ station_id: ObjectIdType
GbfsSystemAlert
+ type: GbfsAlertTypeEnum
+ times: GbfsAlertTimebands
+ station_ids: ObjectIdType [0..*]
+ url: anyUrl [2..1]
+ summary: string
+ description: string [0..1]
+ last_reported: dateTime [0..1]
«PK»
+ alert_id: ObjectIdType [0..1]
«FK»
+ region_ids: ObjectIdType [0..*]
GbfsNetworkModel:: GbfsRentalApp
+ store_uri: anyUrl
+ discovery_uri: anyUrl [0..1]
+gire with
0..*
+classified by
0..*
+applies during
+usable by
0..1
+classifying
0..1
+applies to
0..*
GbfsFaresModel:: GbfsSystemPricingPlan
GbfsNetworkModel:: GbfVehicleType
+providing
0..*
0..*
+available at
GbfsVehicleTypesAvailable
+ count: integer
«FK»
+ vehicle_type_id: ObjectIdType
«enumeration» GbfsValues:: GbfsAlertTypeEnum
SYSTEM_CLOSURE
STATION_CLOSURE
STATION_MOVE
OTHER
GbfsAlertTimeband
+ end_time: dateTime
+ start_time: dateTime

Figure 4 — **GBFS Real time Status Model (UML)**

A.2 GBFS outline mapping to NeTEx and SIRI

The static reference elements of the GBFS model can be mapped to NeTEx equivalents. In a few cases the mapping is one-to-one, e.g., a **GBFS System region** can be represented by a NeTEx TOPOGRAPHIC PLACE. However, in most cases Transmodel/NeTEx breaks concepts down into separate, more normalised, entities. Also, most NeTEx elements are specialisations of abstract elements that provide common properties.

In particular it should be noted that a GBFS **System Information** is in effect a view that combines several separate concepts, corresponding in Transmodel/NeTEx thus to all of (i) a specialisation of a MOBILITY SERVICE (usually a VEHICLE SHARING SERVICE); (ii) An OPERATOR who runs the service, with CONTACT details; (iii) A specific FARE PRODUCT representing the "GBFS plan" that a user of the system may purchase, with URLs to download mobile apps for the product (NeTEx INFOLINKs); (iv) One or more VERSION FRAMEs that specifies the validity of the data set. Elements.

The NeTEx representation of the GBFS **Pricing plan** uses the normal NeTEx fare description elements which allows for multiple FARE PRODUCTS, different user types prices that are separate from the product description, etc., etc. NeTEx allows specific access rights to be assigned to a product, including stepped charge rates, equivalent to a GBFS kilometre and time based rates. A NeTEx representation can also explicitly describe the spatial region of operation as a FARE ZONE. NeTEx can also describe how to purchase and pay for usage of the product and any media proof of purchase as part of a SALES OFFER PACKAGE.

A high level mapping of GBFS elements to NeTEx is shown in the following table. A full attribute level mapping and examples are shown later below.

Table 1 — High level mapping of GBFS to Transmodel/NeTEx

GBFS record .json	**Scope**	**Transmodel / NeTEx**
GBFS_feed	Feed source	DATA SOURCE, VERSION FRAME
system_information	System information including operator, System location, year implemented, URLs, contact info, time zone.	MOBILITY SERVICE, OPERATOR, CONTACT, LOCALE, FARE PRODUCT, VERSION FRAME, (Also NETWORK, FARE ZONE, SALES OFFER PACKAGE),
	Application URLs	INFOLINK
station_information	Static list of all stations, their capacities and locations..	STOP PLACE, PARKING, PARKING AREA, . PARKING CAPACITY

station_status	Number of available bikes and docks at each station and station availability	MONITORED PARKING BAY, PARKING BAY STATUS. (See SIRI for real-time) For historic values see RENTAL AVAILABILITY, PARKING BAY CONDITION
system_hours	Hours of operation for the system	DAY TYPE, PROPERTY OF DAY, TIME BAND, (Also, AVAILABILITY CONDITION)
system_calendar.	Days of operation for the system	SERVICE CALENDAR, DAY TYPE ASSIGNMENT
free_bike_status	Bikes that are available for rent. Required of systems that don't utilize docks or offer bikes for rent outside of stations.	VEHICLE (See SIRI for real-time)
system_regions	The regions into which the system is broken up into	TOPOGRAPHIC PLACE, (Also FARE ZONE, MOBILITY CONSTRAINT ZONE)
system_pricing_plans	Pricing for the system.	FARE PRODUCT, FARE PRICE. TIME INTERVAL TIME INTERVAL PRICE GEOGRAPHICAL INTERVAL, GEOGRAPHICAL INTERVAL PRICE PRICING RULE (Also, TARIFF, FARE STRUCTURE ELEMENT, , FARE ZONE),
system_alerts	Current system alerts	NOTICE, NOTICE ASSIGNMENT (See SIRI for real-time). A Transmodel has MESSAGE provides an equivalent concept

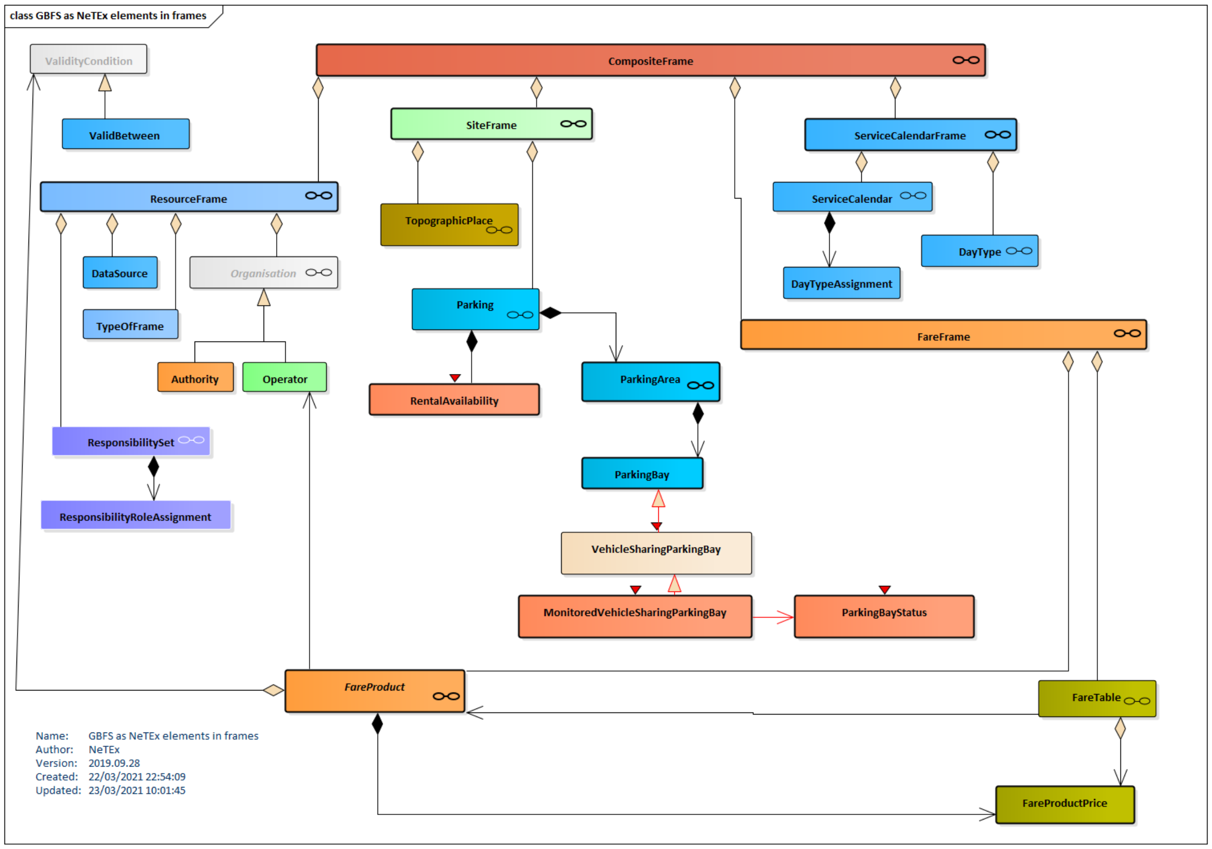
Three of the GBFS services are used to exchange real-time status of hire locations and vehicles, as covered by the SIRI Facility Monitoring service (SIRI-FM). All other GBFS data sets (i.e., *system_information.json*, *station_information.json*, *vehicle_types.json*, *geofencing_zones.json*, etc.) are mapped to NeTEx. The following table shows the mapping of services.

Table 2 —High level mapping of GBFS to Transmodel/SIRI

GBFS record .json	**Siri Service**	**Transmodel Concepts involved**
station_status	SIRI Facility Monitoring	PARKING + status
free_bike_status	SIRI Facility Monitoring	VEHICLE +
system_alerts	SIRI Situation Exchange	SITUATION

A.2.1 NeTEx elements needed to represent GBFS

The equivalent NeTEx elements to represent the various static elements of a GBFS feed are shown in the following figure, grouped within VERSION FRAMES.



A.3 GBFS to NeTEx Mapping tables

The following table shows an attribute level mapping of GBFS data structures to NeTEx XML elements:

Note all NeTEx entities are grouped within a specific version frame, e.g., ***SiteFrame, MobilityServiceFrame***, FareFrame, etc. the Frames are themselves be grouped with in a ***CompositeFrame.***

A.3.1 Table 1 – gbfs.json – GBFS to NeTEx Mapping Table

gbfs.json: Auto-discovery file that links to all of the other files published by the system. The gbfs.json discovery file should represent a single system or geographic area in which vehicles are operated. The location (URL) of the gbfs.json file should be made available to the public using the specification's auto-discovery function.

GBFS Arrays/Fields	Description	Required/ Optional/	NeTEx correspondence indication; comments	Corresponding NeTEx class/attribute
language	The language that will be used throughout the rest of the files.	Required	Version Frame has a default language to be used for all elements that do not specify one	***CompositeFrameDefaults.DefaultLocale.DefaultLanguage***
feeds	An array of all the feeds that are published by this auto-discovery file. Each element in the array is an object with the keys below (name, url).	Required	sequence of VERSION FRAMEs & ***DataSources***	***ResourceFrame,dataSources***
--name	Key identifying the type of feed this is. The key must be the base file name defined in the spec for the corresponding feed type (system_information for system_information.json file, station_information for station_information.json file).	Required	Inherited by ***DataSource*** from ***TypeOfValue***	**DataSource.Version.Name**
--url	URL for the feed. Note that the actual feed endpoints (urls) may not be defined in the *file_name.json* format. For example, a valid feed endpoint could end with **station_information** instead of station_information.json.	Required	Inherited by ***DataSource*** from ***TypeOfValue***	**DataSource.Version.Url**

A.3.2 Table 3 – gbfs_versions.json – GBFS to NeTEx Mapping Table

gbfs_versions.json: Lists all feed endpoints published according to versions of the GBFS documentation.

GBFS Arrays/Fields	Description	Required/ Optional/ Conditional	NeTEx correspondence indication; comments	Corresponding NeTEx class/attribute

versions	Contains one object, as defined below, for each of the available versions of a feed. The array MUST be sorted by increasing MAJOR and MINOR version number.	Required		
::version	The semantic version of the feed in the form X.Y.	Required	***ResourceFrame,dataSources***	***DataSource.Version***
::url	URL of the corresponding *gbfs.json* endpoint.	Required	***ResourceFrame,dataSources***	***DataSource.Url***

A.3.3 Table 4 – gbfs_system_information.json – GBFS to NeTEx Mapping Table

system_information.json: Details including system operator, system location, year implemented, URL, contact info, time zone.

GBFS Arrays/Fields	***Description***	***Required / Optional***	***NeTEx correspondence indication; comments***	***Corresponding NeTEx class/attribute***
system_id	Identifier for this vehicle share system. This should be globally unique (even between different systems) - for example, *bicycle_austin* or *biketown_pdx*. It is up to the publisher of the feed to guarantee uniqueness. This value is intended to remain the same over the life of the system.	Required	***MobilityServiceFrame.mobilityServices***	***VehicleSharingService.id***
language	The language that will be used throughout the rest of the files. It must match the value in the gbfs.json file.	Required	Set a default language on outermost frame.	***VersionFrame.FrameDefaults.DefaultLocale.DefaultLanguage***
name	Name of the system to be displayed to customers.	Required	***MobilityServiceFrame.mobilityServices***	***VehicleSharingService.Name***
short_name	Optional abbreviation for a system.	Optional	***MobilityServiceFrame.mobilityServices***	***VehicleSharingService.ShortName***
operator	Name of the operator.	Optional	***ResourceFrame.operators***	***Operator.Name or Operator.BrandName***
url	The URL of the vehicle share system.	Optional	***ResourceFrame.operators***	***Operator.ContactDetails.Url***
purchase_url	URL where a customer can purchase a membership.	Optional	***FareFrame,distributionChannels***	***DistributionChannel.Url***
start_date	Date that the system began operations.	Optional	***MobilityServiceFrame.mobilityServices***	***VehicleSharingService.StartDate***

phone_number	A single voice telephone number for the specified system that presents the telephone number as typical for the system's service area. It can and should contain punctuation marks to group the digits of the number	Optional	***ResourceFrame.operators***	***Operator.ContactDetails.Phone***
email	Email address actively monitored by the operator's customer service department. This email address should be a direct contact point where riders can reach a customer service representative.	Optional	***ResourceFrame.operators***	***Operator.ContactDetails.Email***
feed-contact_email	A single contact email address for consumers of this feed to report technical issues.	Optional	***ResourceFrame.dataSources***	***DataSource.Email***
timezone	The time zone where the system is located.	Required	***MobilityServiceFrame***	*VersionFrame.FrameDefaults.DefaultLocale.**Locale.TimeZone***
license_id	Required if the dataset is provided under a standard license. An identifier for a standard license from the SPDX License List. Provide license_id rather than license_url if the license is included in the SPDX License List.	Conditionally required	***ResourceFrame.dataSources***	***dataSources.DataSource.DataLicenseCode***
license_url	Required if the dataset is provided under a customized license. A fully qualified URL of a page that defines the license terms for the GBFS data for this system. Do not specify a license_url if license_id is specified.	Conditionally required	As above	***dataSources.DataSource.DataLicenseUrl***
attribution_organization_name	If the feed license requires attribution, name of the organization to which attribution should be provided.	Optional	***ResourceFrame.typeOfValue***	***Branding.Name***
attribution_url	URL of the organization to which attribution should be provided.	Optional	As above	***Branding.Url***
rental_apps	Contains rental app information in the android and ios JSON objects.	Optional	***MobilityServiceFrame.mobilityServices***	***VehicleSharingService.infolinks.***
-android	Contains rental app download and app discovery information for the Android platform in the store_uri and discovery_uri fields.	Optional	As above	
store_uri	URI where the rental Android app can be downloaded from.	Conditionally required	As above	***VehicleSharingServices.infoLinks.Infolink.(****typeOfInfoLink=mobileAppDownload, targetPlatform=android****)***
discovery_uri	URI that can be used to discover if the rental Android app is installed on the device.	Conditionally required	As above	***VehicleSharingServices.infoLinks.Infolink.(****typeOfInfoLink=mobileAppInstallCheck, targetPlatform=android)*
ios	Contains rental information for the iOS platform in the store_uri and discovery_uri fields.	Optional		

store_uri	URI where the rental iOS app can be downloaded from.	Conditionally required		***mobilityServices.VehicleSharingServices.infoLinks.Infolink.****(typeOfInfoLink=mobileAppDownload, targetPlatform=ios)*
discovery_uri	URI that can be used to discover if the rental iOS app is installed on the device.	Conditionally required		***mobilityServices.VehicleSharingServices.infoLinks.Infolink.****(typeOfInfoLink=mobileAppInstallCheck, targetPlatform=ios)*

A.3.4 Table 5 – vehicle_types.json – Mapping Table

vehicle_types.json: Describes the types of vehicles that System operator has available for rent. Required of systems that include information about vehicle types in the free_bike_status file. If this file is not included, then all vehicles in the feed are assumed to be non-motorized bicycles.

GBFS Arrays/Fields	***Description***	***Required/ Optional***	***NeTEx correspondence indication; comments***	***Corresponding NeTEx class/attribute***
vehicle_types	Array that contains one object per vehicle type in the system as defined below.	Required	***ResourceFrame.vehicleTypes***	***SimpleVehicleType***
-vehicle_type_id	Unique identifier of a vehicle type.	Required		***SimpleVehicleType.id***
::form_factor	The vehicle's general form factor. Current valid values are: bicycle; car; moped; scooter; other.	Required	As above All values from GBFS are included in the ***PersonalVehicleCategoryEnum***	***SimpleVehicleType.VehicleCategory***
::propulsion_type	The primary propulsion type of the vehicle. Current valid values are: human (pedal or foot propulsion); electric_assist (provides power only alongside human propulsion); electric (contains throttle mode with a battery-powered motor); combustion (contains throttle mode with a gas engine-powered motor).	Required	All values from GBFS are included in the P***ersonalVehicleCategoryEnum***	***SimpleVehicleType.PropulsionType***
::max_range_meters	This represents the furthest distance in meters that the vehicle can travel without recharging or refuelling when it has the maximum amount of energy potential (for example, a full battery or full tank of gas).	Conditionally required	As above	***SimpleVehicleType.MaximumRange***
::name	The public name of this vehicle type.	Optional	As above	***SimpleVehicleType.Name***

A.3.5 Table 6 – station_information.json – Mapping Table

station_information.json: List of all stations, their capacities and locations. Required of systems utilizing docks. All stations included in station_information.json are considered public (e.g., can be shown on a map for public use). Any station that is represented in station_information.json MUST have a corresponding entry in station_status.json.

GBFS ***Arrays/Fields***	***Description***	***Required/ Optional***	***NeTEx correspondence indication; comments***	***Corresponding NeTEx class/attribute***
stations	Array that contains one object per station as defined below.	Required	***SiteFrame.parkings***	***Parking***
::station_id	Identifier of a station.	Required		***Parking.id***
::name	The public name of the station for display in maps, digital signage and other text applications.	Required	Inherits from ***DataManagedObject***	***Parking.Name***
::short_name	Short name or other type of identifier.	Optional	Inherits from DataManagedObject	***Parking.ShortName***
::lat	Latitude of the station in decimal degrees. This field SHOULD have a precision of 6 decimal places (0.000001).	Required	inherits from ***Place***	***Parking.Centroid.Location.Latitude***
::lon	Longitude of the station in decimal degrees. This field SHOULD have a precision of 6 decimal places (0.000001).	Inherits ***Site***	inherits from ***Place***	***Parking.Centroid.Location.Longitude***
::address	Address (street number and name) where station is located. This must be a valid address, not a free-form text description. Example: 1234 Main Street.	Optional	Inherits form ***SiteElement***	***Parking.HouseNumber*** + ***Parking.Street***
::cross_street	Cross street or landmark where station is located.	Optional	Inherits form ***SiteElement***	***Parking.Landmark OR Parking.CrossRoad***
::region_id	Identifier of the region where the station is located.	Optional	Inherits from ***AddressablePlace***	***Parking.TopographicPlaceRef***
::post_code	Postal code where station is located.	Optional	Inherits from ***AddressablePlace***	***Parking.PostalAddress.PostCode***
::rental_methods	Payment methods accepted at this station.	Optional	Enum mappings contactlessPaymentCard = PAYPASS, epayDevice = (APPLEPAY, ANDROIDPAY), travelCard = TRANSITCARD (?), epayAccount = ACCOUNTNUMBER (?), mobilePhone = PHONE	***Parking.PaymentMethods***

::is_virtual_station	Is this station a location with or without physical infrastructures (docks)? True - The station is a location without physical infrastructure, defined by a point (lat/lon) and/or station_area. False - The station consists of physical infrastructure (docks). If this field is empty, it means the station consists of physical infrastructure (docks).	Optional		***Parking.ParkingProperties.ParkingVisibility=unmarked***
::station_area	A GeoJSON multipolygon that describes the area of a virtual station. If **station_area** is supplied, then the record describes a virtual station. If lat/lon and station_area are both defined, the lat/lon is the significant coordinate of the station (e.g. dock facility or valet drop-off and pick up point). **The station_area** takes precedence over any **ride_allowed** rules in overlapping geofencing_zones.	Optional	Inherited from Zone NB if multiple polygons needed, use chile ***ParkingAreas,*** each with its own polygon.	***Parking.polygon***
::capacity	Number of total docking points installed at this station, both available and unavailable, regardless of what vehicle types are allowed at each dock. If this is a virtual station defined using the **is_virtual_station** field, this number represents the total number of vehicles of all types that can be parked at the virtual station. If the virtual station is defined by station_area, this is the number that can park within the station area. If lat/lon are defined, this is the number that can park at those coordinates.	Optional		***Parking.TotalCapacity***
::vehicle_capacity	An object used to describe the parking capacity of virtual stations (defined using the **is_virtual_station** field), where each key is a vehicle_type_id as described in vehicle_types.json and the value is a number representing the total number of vehicles of this type that can park within the virtual station. If the virtual station is defined by station_area, this is the number that can park within the station area. If lat/lon is defined, this is the number that can park at those coordinates.	Optional	Parking,parkingProperties.ParkingProperties.spaces.ParkingCapacity	***ParkingCapacity.ParkingVehicleType + ParkingCapacity.NumberOfSpaces***

::::vehicle_type_capacity	An object where each key is a vehicle_type_id as described in vehicle_types.json and the value is a number representing the total docking points installed at this station, both available and unavailable for the specified vehicle type.	Optional	Parking,parkingProperties.ParkingProperties.spaces.ParkingCapacity	***ParkingCapacity.ParkingVehicleType* + *ParkingCapacity.VehicleTypeRef***
::is_valet_station	Are valet services provided at this station? True - Valet services are provided at this station. False - Valet services are not provided at this station. If this field is empty, it is assumed that valet services are not provided at this station. This field's boolean should be set to true during the hours which valet service is provided at the station. Valet service is defined as providing unlimited capacity at a station.	Optional		***Parking.facilities.SiteFacilitySet.CarServiceFacilityList=valetParking***
::rental_uris	Contains rental URIs for Android, iOS, and web in the android, ios, and web fields.	Optional		
::android	URI that can be passed to an Android app with an android.intent.action.VIEW Android intent to support Android Deep Links.	Optional		***Parking.infolinks.Infolink.targetPlatform=android***
::ios	URI that can be used on iOS to launch the rental app for this station.	Optional		***Parking.infolinks.Infolink.targetPlatform=ios***
::web	URL that can be used by a web browser to show more information about renting a vehicle at this station.	Optional		***Parking.infolinks.Infolink.targetPlatform=web***

A.3.6 Table 7 – system_hours.json – Mapping Table

system_hours.json: Hours of operation for the system. This optional file is used to describe hours and days of operation when vehicles are available for rental. If system_hours.json is not published, it indicates that vehicles are available for rental 24 hours a day, 7 days a week.

GBFS Fields/Arrays	***Description***	***Required/ Optional/ Conditional***	***NeTEx correspondence indication; comments***	***Corresponding NeTEx class/attribute***
rental_hours	Array of objects as defined below. The array must contain a minimum of one object identifying hours for every day of the week or a maximum of two for each day of the week objects (one for each user type).	Required	***AvailabilityCondition*** for Parking	Parking.validityConditions.AvailabilityCondition. dayTypes.DayTypeRef

::user_types	An array of member and/or non-member value(s). This indicates that this set of rental hours applies to either members or non-members only.	Required	For UserTypeEnum (member can be used to indicate simple member. Other categories, e.g. non member, are possible using ***UserProfile***	***FareProduct.parameterAssignments.GenericParameterAssignment.Limitations.UserProfile.***
::days	An array of abbreviations (first 3 letters) of English names of the days of the week for which this object applies (e.g. ["mon", "tue", "wed", "thu", "fri", "sat, "sun"]). Rental hours must not be defined more than once for each day and user type.	Required	***ServiceCalendarFrame***.dayTypes Each day type can be described with properties,including whether it is a holiday, etc.	***DayType.Properties.PropertyOfDay.DaysOfWeek (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday)***
::start_time	Start time for the hours of operation of the system in the time zone indicated in system_information.json.	Required		***DayType.timebands.Timeband.StartTime***
::end_time	End time for the hours of operation of the system in the time zone indicated in system_information.json.	Required		***DayType.timebands.Timeband.EndTime***

A.3.7 Table 8 – system_calendar.json – Mapping Table

system_calendar.json: Describes the operating calendar for a system. This optional file should be published by systems that operate seasonally or do not offer continuous year-round service.

GBFS Fields/Arrays	***Description***	***Required/ Optional/ Conditional***	***NeTEx correspondence indication; comments***	***Corresponding NeTEx class/attribute***
calendars	Array of objects describing the system operational calendar. A minimum of one calendar object is required. If start and end dates are the same every year, then **start_year** and **end_yea**r should be omitted.	Required		***ServiceCalendar***
::start_month	Starting month for the system operations (1-12).	Required		***ServiceCalendar.FromDate***
::start_day	Starting date for the system operations (1-31).	Required		***ServiceCalendar.FromDate***
::start_year	Starting year for the system operations.	Optional		***ServiceCalendar.FromDate***
::end_month	Ending month for the system operations (1-12).	Required		***ServiceCalendar.ToDate***
::end_day	Ending date for the system operations (1-31).	Required		***ServiceCalendar.ToDate***
::end_year	Ending year for the system operations.	Optional		***ServiceCalendar.ToDate***

A.3.8 Table 9 – system_pricing_plans.json – Mapping Table

system_pricing_plans.json: Describes pricing for the system.

GBFS Fields/Arrays	*Description*	*Required/ Optional*	*NeTEx correspondence indication; comments*	*Corresponding NeTEx class/attribute*
plans	Array of objects as defined below.	Required		
::plan_id	Identifier for a pricing plan in the system.	Required	***FareFrame.fareProducts***	***FareProduct.id***
::url	URL where the customer can learn more about this pricing plan.	Optional	As above	***FareProduct.url***
::name	Name of this pricing plan.	Required	AsAbove	***FareProduct.name***
::currency	Currency used to pay the fare. This pricing is in ISO 4217 code: http://en.wikipedia.org/wiki/ISO_4217 (e.g. CAD for Canadian dollars, EUR for euros, or JPY for Japanese yen.)	Required	***FareFrame***	***FareFrame.FrameDefaults.DefaultCurrency***
::price	Fare price, in the unit specified by currency. If string, must be in decimal monetary value (added in v2.1-RC2). Note: v3.0 may only allow non-negative float, therefore future implementations should be non-negative float. In case of non-rate price, this field is the total price. In case of rate price, this field is the base price that is charged only once per trip (e.g., price for unlocking) in addition to per_km_pricing and/or per_min_pricing.	Required	***FareFrame***	FarePrice.Amount
::is_taxable	Will additional tax be added to the base price? True - Yes. False - No. False may be used to indicate that tax is not charged or that tax is included in the base price.	Required	***FareFrame.PricingParameters.pricingRules*** NB tax rate =can be given	***PricingRule.id=e="gbfs_tax"*** + ***PricingRule.methodName="tax"***
::description	Customer-readable description of the pricing plan. This should include the duration, price, conditions, etc. that the publisher would like users to see.	Required	***FareFrame.tariffs***	FareProduct.description (Version Frame)
::*per_km_pricing	(added in v2.1-RC2) Array of segments when the price is a function of distance travelled, displayed in kilometers. Total price is the addition of price and all segments in per_km_pricing and per_min_pricing.	Optional	***FareFrame.tariffs.Tariff.geographicalIntervals***	***GeographicalInterval***

	If this array is not provided, there are no variable prices based on distance.		***FareFrame.prices***	***GeographicalIntervalPrice***
::::start	(added in v2.1-RC2) Number of kilometers that have to elapse before this segment starts applying.	Required		***GeographicalInterval.StartGeographicalValue***
::::rate	(added in v2.1-RC2) Rate that is charged for each kilometer interval after the start. Can be a negative number, which indicates that the traveller will receive a discount.	Required	***FareFrame.prices***	***GeographicalIntervalPrice.Amount***
::::interval	(added in v2.1-RC2) Interval in kilometers at which the rate of this segment is either reapplied indefinitely, or if defined, up until (but not including) end kilometer. An interval of 0 indicates the rate is only charged once.	Required	***FareFrame.tariffs.Tariff.geographicalIntervals***	***GeographicalInterval.NumberOfUnits***
::::end	(added in v2.1-RC2) The kilometer at which the rate will no longer apply. If this field is empty, the price issued for this segment is charged until the trip ends, in addition to following segments.	Optional	As above	***GeographicalInterval.EndGeographicalValue***
::*per_min_pricing	(added in v2.1-RC2) Array of segments when the price is a function of time travelled, displayed in minutes. Total price is the addition of price and all segments in per_km_pricing and per_min_pricing. If this array is not provided, there are no variable prices based on time.	Optional	***FareFrame.tariffs.Tariff.timeIntervals***	***TimeInterval***
			FareFrame.prices	***TimeIntervalPrice***
::::start	(added in v2.1-RC2) Number of minutes that have to elapse before this segment starts applying.	Required	***FareFrame.tariffs.Tariff.timeIntervals***	***TimeInterval.StartTime***
::::rate	(added in v2.1-RC2) Rate that is charged for each minute interval after the start. Can be a negative number, which indicates that the traveller will receive a discount.	Required	As above	***TimeIntervalPrice.Amount***
::::interval	(added in v2.1-RC2) Interval in minutes at which the rate of this segment is either reapplied indefinitely, or if defined, up until (but not including) end minute. An interval of 0 indicates the rate is only charged once.	Required	As above	***TimeInterval.Duration***
::::end	(added in v2.1-RC2) The minute at which the rate will no longer apply. If this field is empty, the price issued for this segment is charged until the trip ends, in addition to following segments.	Optional	As above	***TimeInterval.EndTime***

::surge_pricing	(added in v2.1-RC2) Is there currently an increase in price in response to increased demand in this pricing plan? If this field is empty, it means these is no surge pricing in effect. true - Surge pricing is in effect. false - Surge pricing is not in effect.	Optional	***FareFrame.PricingParameters.pricingRules*** NB surge rate =can be given	***PricingRule.id=”***gbfs.surge” + ***PricingRule.methodName=”surge”***

A.3.9 Table 10 – geofencing_zones.json – Mapping Table

geofencing_zones.json: Geofencing zones and their associated rules and attributes. Describes geofencing zones and their associated rules and attributes. By default, no restrictions apply everywhere. Geofencing zones should be modelled according to restrictions rather than allowance. An operational area (outside of which vehicles cannot be used) should be defined with a counter-clockwise polygon, and a limitation area (in which vehicles can be used under certain restrictions) should be defined with a clockwise polygon.

GBFS Fields/Arrays	***Description***	***Required/ Optional***	***NeTEx correspondence indication; comments***	***Corresponding NeTEx class/attribute***
geofencing_zones	Each geofenced zone and its associated rules and attributes is described as an object within the array of features, as follows.	Required	***MobilityServiceFrame.mobilityServiceConstraintZone***	***MobilityServiceConstraintZone***
type	“FeatureCollection” (as per IETF RFC 7946).	Required		***MobilityServiceConstraintZone.Types.TypeOfZoneRef***
***features**	Array of objects as defined below.	Required		
::type	“Feature” (as per IETF RFC 7946).	Required		SIMPLE FEATURE
::::geometry	A polygon that describes where rides might not be able to start, end, go through, or have other limitations. A clockwise arrangement of points defines the area enclosed by the polygon, while a counter clockwise order defines the area outside the polygon (right-hand rule). All geofencing zones contained in this list are public (i.e., can be shown on a map for public use).	Required	***Zone*** geometry defined by group of points Inside /outside explicity attribute	***MobilityService ConstraintZone,polygon*** ***MobilityServiceConstraintZone. RuleApplicability***=i*”nside”| “outside”)
::::properties	Properties: As defined below, describing travel allowances and limitations.	Required	MobilityServiceFrame.mobilityServiceConstraintZones.MobilityServiceConstraintZone.validityConditions	AvailabilityCondition
::::::name	Public name of the geofencing zone.	Optional		***MobilityServiceConstraintZone.Name***
::::::start	Start time of the geofencing zone. If the geofencing zone is always active, this can be omitted.	Optional	***MobilityServiceConstraintZone.validityConditions***	***AvailabilityCondition.FromDate***

::::::end	End time of the geofencing zone. If the geofencing zone is always active, this can be omitted.	Optional	As above	***AvailabilityCondition.ToDate***
::::::rules	Array that contains one object per rule as defined below. In the event of colliding rules within the same polygon, the earlier rule (in order of the JSON file) takes precedence. In the case of overlapping polygons, the combined set of rules associated with the overlapping polygons applies to the union of the polygons. In the event of colliding rules in this set, the earlier rule (in order of the JSON file) also takes precedence.	Optional	***MobilityServiceConstraintZone.vehicleRestrictions***	***VehicleTypeZoneRestriction***
::::::::vehicle_type_id	Array of IDs of vehicle types for which any restrictions should be applied (see vehicle type definitions in PR #136). If vehicle_type_ids are not specified, then restrictions apply to all vehicle types.	Optional	As above	***VehicleTypeZoneRestriction.SimpleVehicleTypeRef***
::::::::ride_allowed	Is the undocked ("free bike") ride allowed to start and end in this zone? true - Undocked ("free bike") ride can start and end in this zone. false - Undocked ("free bike") ride cannot start and end in this zone.	Required	As above	***VehicleTypeZoneRestriction.ZoneUse***=allUsesAllowed OR ***ZoneUse***=forbiddenZone)
::::::::ride_through_allowed	Is the ride allowed to travel through this zone? true - Ride can travel through this zone. false - Ride cannot travel through this zone.	Required	As above	**VehicleTypeZoneRestriction.ZoneUse**=passThroughUseOnly OR **ZoneUse**=noPassThrough)
::::::::maximum_speed_kph	What is the maximum speed allowed, in kilometers per hour? If there is no maximum speed to observe, this can be omitted.	Optional	As above	***VehicleTypeZoneRestriction.MaximumSpeed***

A.4 GBFS to SIRI Mapping tables

A.4.1 Table 11 – station_status.json – GBFS to SIRI mapping table.

station_status.json: Number of available vehicles and docks at each station and station availability. Required of systems utilizing docks. Describes the capacity and rental availability of a station.

GBFS Arrays/Fields	***Description***	***Required/ Optional/ Conditional***	***SIRI correspondence indication; comments***	***Corresponding SIRI class/attribute***
stations	Array that contains one object per station in the system as defined below.	Required	SIRI Facility Monitoring service	***FacilityMonitoringDelivery***
::station_id	*Identifier of a station see station_information.json.*	Required	Station Id corresponds to NeTEx value	***FacilityCondition.FacilityRef***=Station.Id (reference to a station described used NeTEx)
::num_bikes_available	Number of vehicles of any type available for rental. Number of functional vehicles physically at the station. To know if the vehicles are available for rental, see **is_renting.**	Required	**SIRI FacilityCondition and MonitoredCounting**	SIRI **FacilityCondition.FacilityRef**=Station.Id **FacilityCondition.MonitoredCounting** ::**CountingType**=*availabilityCount* ::**CountedFeatureUnit**=*vehicles* ::**Count**=num_bikes_available
::vehicle_types_available	This field is required if the vehicle_types.json file has been defined. This field's value is an array of objects. Each of these objects is used to model the total number of each defined vehicle type available at a station. The total number of vehicles from each of these objects should add up to match the value specified in the **num_bikes_available** field**.**	Conditionally required	SIRI **FacilityCondition** and **MonitoredCounting**	Same as above but with a **TypeOfCountedFeature.TypeOfValueCode**=id of the VehicleType Then the MonitoredCounting block has to be repeated to each VehicleType If the list of Ids of Vehicles is provided, An alternative is not to provide the ***VehicleType*** in the SIRI flow but to relay on the fact that in the NeTEx data set each Vehicle has an associated ***VehicleType***.
::vehicle_type_id	The vehicle_type_id of each vehicle type at the station as described in vehicle_types.json. This field is required if the *vehicle_types.json* is defined.	Required	NeTEx and SIRI	***SimpleVehicleType.id*** in NeTEx or see above for the SIRI realtime part
::count	A number representing the total number of available vehicles of the corresponding **vehicle_type_id** as defined in *vehicle_types.json* at the station.	Required	SIRI **FacilityCondition** and **MonitoredCounting**	(see above) ***FacilityCondition.MonitoredCounting*** ::***CountingType***=*availabilityCount* ::***CountedFeatureUnit***=*vehicles* TypeOfCountedFeature.TypeOfValueCode=vehicle_type_id ::Count=num_bikes_available

::num_bikes_disabled	Number of disabled vehicles of any type at the station. Vendors who do not want to publicize the number of disabled vehicles or docks in their system can opt to omit station capacity (in station_information), num_bikes_disabled and num_docks_disabled (as of v2.0). If station capacity is published, then broken docks/vehicles can be inferred (though not specifically whether the decreased capacity is a broken vehicle or dock).	Optional	SIRI **FacilityCondition** and **MonitoredCounting**	***FacilityCondition.MonitoredCounting*** ::***CountingType***=outOfOrderCount ::***CountedFeatureUnit***=*vehicles* *::**Count***=num_bikes_disabled
::num_docks_available	Required except for stations that have unlimited docking capacity (e.g. virtual stations) (as of v2.0). Number of functional docks physically at the station. To know if the docks are accepting vehicle returns, see is_returning.	Conditionally required	SIRI **FacilityCondition** and **MonitoredCounting**	***FacilityCondition.MonitoredCounting*** ***CountingType***=availabilityCount ***CountedFeatureUnit***=bays ***Count***=num_docks_available
::vehicle_docks_available	This field is required in feeds where the vehicle_types.json is defined and where certain docks are only able to accept certain vehicle types. If every dock at the station is able to accept any vehicle type, then this field is not required. This field's value is an array of objects. Each of these objects is used to model the number of docks available for certain vehicle types. The total number of docks from each of these objects should add up to match the value specified in the num_docks_available field.	Conditionally required	SIRI **FacilityCondition** and **MonitoredCounting**	***FacilityCondition.MonitoredCounting*** ***CountingType***=*availabilityCount* ***CountedFeatureUnit***=bays ***TypeOfCountedFeature.TypeOfValueCode***=vehicle_type_id ***Count***=num_docks_available
vehicle_type_ids	An array of strings where each string represents a vehicle_type_id that is able to use a particular type of dock at the station.	Required	FacilityCondition and MonitoredCounting CountedFeatureUnit and TypeOfCountedFeature	***SimpleVehicleType.id*** in NeTEx for the static description or see above in SIRI
count	A number representing the total number of available vehicles of the corresponding vehicle type as defined in the **vehicle_types** array at the station that can accept vehicles of the specified types in the v**ehicle_types** array.	Required		***FacilityCondition.MonitoredCounting*** ::***CountingType***=*availabilityCount* ::***CountedFeatureUnit***=bays ::***TypeOfCountedFeature***.TypeOfValueCode=vehicle_type_id :: Count=num_docks_available

num_docks_disabled	Number of empty but disabled dock points at the station.	Optional	FacilityCondition and MonitoredCounting	***FacilityCondition.MonitoredCounting*** CountingType=outOfOrderCount CountedFeatureUnit=bays Count=num_docks_disabled
::is_installed	Is the station currently on the street? true - Station is installed on the street. false - Station is not installed on the street. Boolean should be set to true when equipment is present on the street. In seasonal systems where equipment is removed during winter, boolean should be set to false during the off season. May also be set to false to indicate planned (future) stations which have not yet been installed.	Required	NeTEx and SIRI: planned availability is expected to be in NeTEX	For realtime Status ***FacilityCondition.FacilityStatus.Status*** ("*available","notAvailable","partiallyAvailable","added","removed*") Possibly complemented by ***FacilityCondition.FacilityStatus.Description*** Note that a Parking that is not installed anywhere would not appear in the SIRI data flow (see NeTEx availability conditions for more) Use NeTEx ***Parking.AvailabilityCondition*** for planed availability
::is_renting	Is the station currently renting vehicles? true - Station is renting vehicles. Even if the station is empty, if it is set to allow rentals this value should be true. false - Station is not renting vehicles.	Required	FacilityCondition	***FacilityCondition.FacilityStatus***.Status ("available","notAvailable","partiallyAvailable","added","removed") Note that SIRI does not differentiate renting/returning except by providing a number of available bays/vehicle
::is_returning	Is the station accepting vehicle returns? *true* - Station is accepting vehicle returns. If a station is full but would allow a return if it was not full, then this value should be *true*. *false* - Station is not accepting vehicle returns.	Required	FacilityCondition	***FacilityCondition.FacilityStatus***.Status ("*available","notAvailable","partiallyAvailable","added","removed")* Possibly complemented by ***FacilityCondition.FacilityStatus.Description*** Note that SIRI does not differentiate renting/returning except by providing a number of available bays/vehicle
::last_reported	The last time this station reported its status to the operator backend.	Required	FacilityCondition	***FacilityCondition.ValidityPeriod.StartTime***

A.4.2 Table 12 – free_bike_status.json – GBFS to SIRI mapping table

free_bike_status.json: Describes all vehicles that are not currently in active rental. Required for free floating (dockless) vehicles. Optional for station based (docked) vehicles. Vehicles that are part of an active rental must not appear in this feed.

Note: Floating vehicles attribute is contained in the MobilityServiceFrame.mobilityServices.VehicleSharingService.FloatingVehicles (true/false).

GBFS ***Arrays/Fields***	***Description***	***Required/ Optional/ Conditional***	***SIRI correspondence indication; comments***	***Corresponding SIRI class/attribute***
bikes	Array that contains one object per vehicle that is currently stopped as defined below.	Required	Should correspond to data from MobilityServiceFrame.fleets.Fleet.members.vehicle	***FacilityMonitoringDelivery***
::bike_id	Identifier of a vehicle, rotated to a random string, at minimum, after each trip to protect privacy (as of v2.0). Note: Persistent bike_id, published publicly, could pose a threat to individual traveller privacy.	Required	MobilityServiceFrame.mobilityServices.VehicleSharingService.fleets.FleetRef contains reference to the MobilityServiceFrame where Vehicles for the fleet are described.	NeTEx: ***Vehicle.Vehicle.id*** SIRI: ***FacilityCondition.FacilityRef***=bike_id
::lat	Latitude of the vehicle. This field is required if station_id is not provided for this vehicle (free floating).	Conditionally required	FacilityCondition and FacilityUpdatedPosition	***FacilityCondition.FacilityRef***=bike_id ***FacilityCondition.FacilityUpdatedPosition*** ::latitude=lat
::lon	Longitude of the vehicle. This field is required if **station_id** is not provided for this vehicle (free floating).	Conditionally required	FacilityCondition and FacilityUpdatedPosition	**FacilityCondition.FacilityRef**=bike_id **FacilityCondition.FacilityUpdatedPosition** ::longitude=lon
::is_reserved	Is the vehicle currently reserved? *true* - Vehicle is currently reserved. *false* - Vehicle is not currently reserved.	Required	FacilityCondition	***FacilityCondition.FacilityRef***=bike_id ***FacilityCondition.MonitoredCounting*** ::***CountingType***=outOfOrderCount ::***CountedFeatureUnit***=vehicles ::***Count***=1 Alternatively:

				FacilityCondition.FacilityStatus.Status ="available"/"notAvailable"
::is_disabled	Is the vehicle currently disabled? true - Vehicle is currently disabled. false - Vehicle is not currently disabled.	Optional		***FacilityCondition.FacilityRef***=bike_id FacilityCondition.MonitoredCounting ::CountingType=outOfOrderCount ::CountedFeatureUnit=vehicles ::Count=1 Alternatively: FacilityCondition.FacilityStatus.Status ="available"/"notAvailable"
::*rental_uris	JSON object that contains rental URIs for Android, iOS, and web in the android, ios, and web fields.	Optional		***Parking*.infoLinks**
::::android		Optional	From NeTEx	***Parking*.infoLinks.Infolink**.(typeOfInfoLink=info targetPlatform=android)
::::ios		Optional	From NeTEx	***Parking.infoLinks.Infolink***.(typeOfInfoLink=info targetPlatform=ios)
::::web		Optional	From NeTEx	***Parking.infoLinks.Infolink.(***typeOfInfoLink=info targetPlatform=web)
::vehicle_type_id	The **vehicle_type_id** of this vehicle as described in *vehicle_types.json*. This field is required if the *vehicle_types.json* is defined.	Conditionally required	From NeTEx	VehicleType.id
::current_range_meters	If the corresponding **vehicle_type** definition for this vehicle has a motor, then this field is required. This value represents the furthest distance in meters that the vehicle can travel without recharging or refuelling with the vehicle's current charge or fuel.	Conditionally required		***FacilityCondition.FacilityRef***=bike_id ***FacilityCondition.MonitoredCounting*** ::***CountingType***=availableRunningDistance ::***CountedFeatureUnit***=meters ::***Count***=current_range_meters Note that SIRI also provides a ***ChargingLevel*** if necessary.
::station_id	Identifier referencing the **station_id** field in *system_information.json*.	Conditionally required	Bikes at a station are reported in the FacilityCondition of the station	For the station ***FacilityCondition.FacilityRef***=Station.Id ***FacilityCondition.MonitoredCounting,***

				CountedItemsIdList for the list of vehicles
::pricing_plan_id	The **plan_id** of the pricing plan this vehicle is eligible for as described in *system_pricing_plans.json.*	Optional	From NeTEx	***FareProduct.id***

A.4.3 Table 13 – system_alerts.json – GBFS to SIRI mapping table.

The mapping of the GBFS *system_alerts.json* is not provided as a table: the system alerts map to the Situation Exchange SIRI Service (SX), and an example of conversion of a GBFS system alert to SIRI SX is provided below.

A.5 GBFS to SIRI Comments respecting the mapping table

A.5.1 Use Installed, Renting and Returning attributes

SIRI does not have elements dedicated to the GBFS *station_status* **is_installed, is_renting** and ***is_returning*** attributes of a however this information can be mapped as follows:

- **Is_installed** is relevant for static information and shall be mapped using the ***ValidityCondition*** (or more likely ***AvailabilityCondition***) provide by NeTEx for Parking (knowing that a GBFS *station* maps to a NeTEx ***Parking***)
- SIRI provides a ***Status*** (*available/notAvailable/partiallyAvailable*) therefore
  - o *A renting* and *returning* station will have a ***Status*** of *available* and will have a positive, non-zero, count for available vehicles and available bays (place to bring back a vehicle)
  - o *A renting* but not *returning* station will have a ***Status*** of *partiallyAvailable* and will have a positive, non-zero, count for available vehicles and no available bays (count=0)
  - o *A returning* but not *renting* station will have a ***Status*** of *partiallyAvailable* and will have a positive, non-zero, count for available bays and no available vehicles (count=0)
  - o A station that is neither *returning* and *renting* (but still in the operating hours as defined by one or more ***AvailabilityCondition*** instances) will have a ***Status*** of *notAvailable* and will have no available bays and no available bays (count=0)

A.5.2 Details of vehicles available at a station

The GBFS *free_bike_status.json* provides information on individual **GBFS bikes** (ie vehicles), including free floating vehicles and vehicles within a station. In SIRI style, detailed information about the vehicles available at a station will have to be done when providing information about that station (see example below). So the *free_bike_status.json* dataset will be split in 2 to be described in SIRI: free floating vehicles in one hand and vehicles at a station in the other. The historic value may be recorded in NeTEx as a ***RentalAvailability.***

A.5.3 Principle of the SIRI MonitoredCounting

Unlike GBFS, which covers just the specific use case of the real-time availability of vehicle sharing, the SIRI Facility Monitoring Service has a much wider functional scope. It can provide the real-time status of any facility, a facility potentially being a site (parking, stop place, POI, etc.), a piece of equipment (lift, escalator, ticket machine, etc.), a service (boarding assistance, guidance service, etc.) or a vehicle (car, bike, bus, train, etc.). This facility status can include quantitative counts (***MonitoredCounting***) such as the number of available vehicles, the number of available bays (place where to park), but also the number of wheel-chairs or walking stick (when applicable) available at a station or at a Point of Interest, or the number of available seats in a vehicle or in a space, the number of persons or even the current temperature.

The GBFS to SIRI Facility Monitoring thus only uses a small part of what can be expressed with SIRI, and any information coming from this mapping can be complemented with further information when needed (and available of course). The mapping given here considers only the GBFS scope (For the full SIRI functional scope see SIRI Part 4, Facility Monitoring, and Part 5, Situation Exchange,).

A.5.4 SIRI open types: TypeOfCountedFeature

SIRI often offers pre-coded values and open-ended values that can be used in combination. This is especially the case in ***MonitoredCounting*** which has both a ***CountedFeatureUnit*** (pre-coded enumeration) and ***TypeOfCountedFeature*** (open-ended type of value) that holds a code (for the GBFS mapping) identifying the precise type of vehicle. The ***VehicleType*** (or ***SimpleVehicleType***) itself is expected to be comprehensively described in a corresponding NeTEx data set.

A.5.5 Data source mention recommendation

A GBFS data set will be mapped as a NeTEx data set complemented by a SIRI Facility Monitoring service, and may also use a SIRI Situation Exchange service for alerts describing disruptions to the service. NeTEx has the ability to provide a ***DataSource*** element: it is highly recommended that this ***DataSource*** provides the information that the data set originate from a GBFS one (this can be simply done using its *Description* element, but also in a more structured way using its ***PrivateCode*** element).

A.6 Examples of Mapping - GBFS

In this section the GBFS json files and the correspondent NeTEx/SIRI mappings are presented.

Note that the source information for GBFS is https://gbfs.mobilitydata.org/specification, but the examples are also based in https://mobilitydata.medium.com/gbfs-now-fully-supports-dockless-systems-289efb6b7c6f.

A.6.1 *gbfs.json* – Example

GBFS file: *gbfs.json*:

```
{
        "last_updated": 1609866247,
        "ttl": 0,
        "version": "2.2",
        "data": {
                "en": {
                        "feeds": [
                                {
                                        "name": "system_information",
                                        "url":
"https://www.example.com/gbfs/1/en/system_information"
                                },
                                {
                                        "name": "station_information",
                                        "url":
"https://www.example.com/gbfs/1/en/station_information"
                                }
                        ]
                },
                "fr" : {
                        "feeds": [
                                {
                                        "name": "system_information",
                                        "url":
"https://www.example.com/gbfs/1/fr/system_information"
                                },
                                {
                                        "name": "station_information",
                                        "url":
"https://www.example.com/gbfs/1/fr/station_information"
                                }
                        ]
                }
        }
}
```

With NeTEx:

```
<ResourceFrame version="1.0" id="rf_01">
        <Name>Specific common information </Name>
        <dataSources>
                <DataSource version="2.0" id="ds_01">
                        <!-- GBFS name -->
                        <Name lang="en">gbfs:system_information</Name>
                        <!-- GBFS url -->
```

```
                    <Url>https://www.example.com/gbfs/1/en/system_information</Url>
                </DataSource>
                <DataSource version="2.0" id="ds_02">
                        <!-- GBFS name -->
                        <Name lang="en"> gbfs:station_information</Name>
                        <!-- GBFS url -->
                        <Url>https://www.example.com/gbfs/1/en/station_information</Url>
                </DataSource>
                <DataSource version="2.0" id="ds_03">
                        <!-- GBFS name -->
                        <Name lang="fr"> gbfs:system_information</Name>
                        <!-- GBFS url -->
                        <Url>https://www.example.com/gbfs/1/fr/system_information</Url>
                </DataSource>
                <DataSource version="2.0" id="ds_04">
                        <!-- GBFS name -->
                        <Name lang="fr"> gbfs:station_information</Name>
                        <!-- GBFS url -->
                        <Url>https://www.example.com/gbfs/1/fr/station_information</Url>
                </DataSource>
        </dataSources>
</ResourceFrame>
```

A.6.2 gbfs_versions.json – Example

GBFS file: gbfs_versions.json:

```
{
        "last_updated": 1609866247,
        "ttl": 0,
        "version": "2.2",
        "data": {
                "versions": [
                        {
                                "version":"2.0",
                                "url":"https://www.example.com/gbfs/2/gbfs"
                        },
                        {
                                "version":"2.2",
                                "url":"https://www.example.com/gbfs/2-2/gbfs"
                        }
                ]
        }
}
```

NeTEx:

```
<frames>
   <!-- ==== COMMON RESOURCES ==== -->
   <ResourceFrame version="1.0" id="rf_01">
      <Name>Vehicle Specific common information </Name>
      <dataSources>
        <DataSource version="2.0" id="ds_01">       <!-- GBFS version -->
                <Url>https://www.example.com/gbfs/2/gbfs</Url> <!-- GBFS url -->
        </DataSource>
        <DataSource version="2.2" id="ds_02"> <!-- GBFS version -->
                <Url>https://www.example.com/gbfs/2-2/gbfs</Url> <!-- GBFS url -->
        </DataSource>
      </dataSources>
   </ResourceFrame>
</frames>
```

A.6.3 *system_information.json* – Example

GBFS file: *system_information.json*:

```
{
        "last_updated":1611598155,
        "ttl":1800,
        "version": "2.2",
        "data":{
                "system_id":"EXM01",
                "language":"en",
                "phone_number":"1-800-555-1234",
                "name":"Example Ride",
                "operator":"Example Sharing, Inc",
                "start_date":"2010-06-10",
                "purchase_url":"https://www.exampleride.org",
                "timezone":"US/Central",
                "license_url":"https://exampleride.org/data-license.html",
                "short_name":"Example Ride",
                "email":"customerservice@exampleride.org",
                "url":"http://www.exampleride.org",
                "feed_contact_email": datafeed@exampleride.org,
                "rental_apps":{
                        "android": {
                                "discovery_url":"coma.bcrent.android//"
                                "store_url":
"https://play.google.com/store/apps/details?id=com.abcrent.android"
                        }
                        "ios": {
                                "discovery_url":"coma.bcrent.ios//"
                                "store_url": "https://apps.apple.com/apps/apple-stire/id123456"
                        }
                }
        }
}
```

With NeTEx:

```
<ResourceFrame version="1.0" id="EXM01">
      <Name>Vehicle sharing system information specific common information </Name>
      <dataSources>

            <DataSource version="any" id=" EXM01">
               <!-- GBFS attribution_organization_name -->
               <Name>feed licence attribution organization name</Name>
               <!-- GBFS attribution_url -->
               <Url>https://exampleride.org/data-license-attribution.html</Url>
               <!-- GBFS feed_contact_email -->
               <Email>datafeed@exampleride.org</Email> <!-- GBFS licence_id -->
               <DataLicenceCode type="SPX" ref="GSCYQ"/> <!-- GBFS licence_url -->
               <DataLicenceUrl>https://exampleride.org/data-license.html</DataLicenceUrl>
            </DataSource>
      </dataSources>

      <typesOfValue>
            <Branding version="any" id="EXM01"> <!-- GBFS attribution_organization_name -->
               <Name>feed licence attribution organization name</Name>
               <!-- GBFS attribution_url -->
               <Url>https://exampleride.org/data-license-attribution.html</Url>
            </Branding>
      </typesOfValue>

      <organisations>
```

```
            <!-- ==== ORGANISATIONS ==== -->
            <Operator version="any" id="EXM01"> <!-- GBFS operator -->
               <Name>Example Sharing, Inc</Name>
               <ContactDetails>
                      <!-- GBFS email -->
                      <Email>customerservice@exampleride.org</Email>
                      <Phone>1-800-555-1234</Phone>
                      <Url>http://www.exampleride.org</Url>
               </ContactDetails>
               <OrganisationType>operator</OrganisationType>
            </Operator>
      </organisations>

      <modesOfOperation>
            <VehicleSharingOperation version="any" id="EXM01">
               <Name>Example Ride</Name>
               <VehicleSharingOperationType>vehicleSharing</VehicleSharingOperationType>
            </VehicleSharingOperation>
      </modesOfOperation>
</ResourceFrame>
<!--==== Mobility Service ====-->
<MobilityServiceFrame version="1.0" id="EXM01">
      <prerequisites>
            <ResourceFrameRef version="1.0" ref="EXM01"/>
      </prerequisites>
      <!--=== Available services ===-->
      <mobilityServices>
            <VehicleSharingService version="any" id="EXM01">
               <Name lang="en">Example Ride</Name> <!-- GBFS name -->
            <infoLinks>
                      <InfoLink typeOfInfoLink="info">coma.bcrent.android//</InfoLink>
                      <InfoLink typeOfInfoLink="info"
                             targetPlatform="ios">coma.bcrent.ios</InfoLink>
                      <InfoLink typeOfInfoLink="mobileAppDownload"

targetPlatform="ios">https://apps.apple.com/apps/apple-stire/id123456</InfoLink>
                      <InfoLink typeOfInfoLink="info"
targetPlatform="android">https://play.google.com/store/apps/details?id=com.abcrent.ios</InfoLin
k>
                      <InfoLink typeOfInfoLink="mobileAppDownload"
                             targetPlatform="android">coma.bcrent.android</InfoLink>
               </infoLinks>
               <ShortName>Example ride</ShortName> <!-- GBFS short_name -->
               <StartDate>2010-06-10</StartDate> <!-- GBFS start_date -->
               <VehicleSharingOperationRef version="any" ref="EXM01"/>
            </VehicleSharingService>
      </mobilityServices>
</MobilityServiceFrame>
```

A.6.4 vehicle_types.json – Example

GBFS file: *vehicle_types.json*:

```
{
       "last_updated": 1609866247,
       "ttl": 0,
       "version": "2.2",
       "data": {
              "vehicle_types": [
                     {
                            "vehicle_type_id": "abc123",
                            "form_factor": "bicycle",
                            "propulsion_type": "human",
                            "name": "Example Basic Bike"
                     },
                     {
```

```
                                "vehicle_type_id": "def456",
                                "form_factor": "scooter",
                                "propulsion_type": "electric",
                                "name": "Example E-scooter V2",
                                "max_range_meters": 12345
                        },
                        {`
                                "vehicle_type_id": "car1",
                                "form_factor": "car",
                                "propulsion_type": "combustion",
                                "name": "Four-door Sedan",
                                "max_range_meters": 523992
                        }
                ]
        }
}
```

With NeTEx:

```
<ResourceFrame version="2.2" id="rf_01">
      <vehicleTypes>
            <SimpleVehicleType version="any" id="abc123">  <!-- GBFS vehicle_type_id -->
               <Name>Example Basic Bike</Name>      <!-- GBFS name -->
               <PropulsionType>human</PropulsionType> <!-- GBFS propulsion_type -->
               <VehicleCategory>cycle</VehicleCategory> <!-- GBFS form_factor -->

            </SimpleVehicleType>
            <SimpleVehicleType version="any" id="def456"> <!-- GBFS vehicle_type_id -->
               <Name>E-scooter</Name> <!-- GBFS name -->
               <PropulsionType>electric </PropulsionType>  <!-- GBFS propulsion_type -->
               <MaximumRange>12345</MaximumRange> <!-- GBFS max_range_meters -->
               <VehicleCategory>scooter</VehicleCategory> <!-- GBFS form_factor -->
            </SimpleVehicleType>
            <SimpleVehicleType version="any" id="car1"> <!-- GBFS vehicle_type_id -->
               <Name>Foor-door Sedan</Name>         <!-- GBFS name -->
               <PropulsionType>combustion</PropulsionType> <!-- GBFS propulsion_type -->
               <MaximumRange>523992</MaximumRange> <!-- GBFS max_range_meters -->
               <VehicleCategory>car</VehicleCategory>      <!-- GBFS form_factor -->
            </SimpleVehicleType>
      </vehicleTypes>
</ResourceFrame>
```

A.6.5 station_information.json – Examples

GBFS file: *station_information.json*:

A.6.5.1 Example 1: Physical station

```
{
        "last_updated": 1609866247,
        "ttl": 0,
        "version": "2.2",
        "data": {
                "stations": [
                        {
                                "station_id": "pga",
                                "name": "Parking garage A",
                                "lat": 12.345678,
                                "lon": 45.678901,
                                "vehicle_type_capacity": {
                                        "abc123": 7,
                                                  “def456": 9
                                    }
                              }
                        ]
            }
}
```

With NeTEx:

```
<SiteFrame version="1.0" id="sf_01">
   <Name>Parking site</Name>
   <parkings>
      <Parking version="any" id="pga">
         <Name lang="en">Parking garage A</Name>
         <Description>Physical station</Description>
         <Centroid>
            <Location id="l_01">
               <Longitude>12.345678</Longitude>
               <Latitude>45.678901</Latitude>
            </Location>
         </Centroid>
         <parkingProperties>
            <ParkingProperties version="any" id="pga">
               <ParkingVehicleTypes>cycle motorScooter</ParkingVehicleTypes>
            </ParkingProperties>
         </parkingProperties>
         <parkingAreas>
            <VehicleSharingParkingArea version="any" id="pga_01">
               <TotalCapacity>7</TotalCapacity>
               <ParkingProperties id="pga_01" version="any">
               <vehicleTypes>
                       <SimpleVehicleTypeRef ref="abc123"/>
               </vehicleTypes>
               </ParkingProperties>
            </VehicleSharingParkingArea>
            <VehicleSharingParkingArea version="any" id="pga_02">

               <TotalCapacity>9</TotalCapacity>
               <ParkingProperties id="pga_02" version="any">
               <vehicleTypes>
                       <SimpleVehicleTypeRef ref="def456"/>
               </vehicleTypes>
               </ParkingProperties>
            </VehicleSharingParkingArea>

         </parkingAreas>
      </Parking>
   </parkings>
</SiteFrame>
```

GBFS file: *station_information.json*:

A.6.5.2 Example 2: Virtual station

```
{
       "last_updated":1609866247,
       "ttl":0,
       "version":"2.2",
       "data":{
              "stations":[
                     {
                            "station_id":"station12",
                            "station_name":"SE Belmont & SE 10 th ",
                            "is_valet_station":false,
                            "is_virtual_station":true,
                                "station_area":{
                                        “type":"MultiPolygon",
                                        “coordinates":[
                                                [
                                                        [
                                                                [
                                                                        -122.655775,
                                                                        45.516445
                                                                ],
                                                                [
                                                                        -122.655705,
                                                                        45.516445
                                                                ],
                                                                [
                                                                        -122.655705,
                                                                        45.516495
                                                                ],
                                                                [
                                                                        -122.655775,
                                                                        45.516495
                                                                ],
                                                                [
                                                                        -122.655775,
                                                                        45.516445
                                                                ]
                                                        ]
                                                ]
                                        ]
                                },
                                "capacity":16,
                                "vehicle_capacity":{
                                        “abc123":8,
                                        “def456":8,
                                        “ghi789":16
                                }
                        }
                ]
        }
}
```

With NeTEx:

```
<ResourceFrame version="1.0" id="EXM02">
   <vehicleTypes>
      <SimpleVehicleType version="any" id="abc123">
         <Name>Bicycle</Name>
         <VehicleCategory>cycle</VehicleCategory>
      </SimpleVehicleType>
      <SimpleVehicleType version="any" id="def456">
         <Name>E-scooter</Name>
         <VehicleCategory>scooter</VehicleCategory>
      </SimpleVehicleType>
      <SimpleVehicleType version="any" id="ghi789">
         <Name>Medium car</Name>
         <VehicleCategory>mediumCar</VehicleCategory>
      </SimpleVehicleType>
   </vehicleTypes>
</ResourceFrame>
<!--==== PARKING (STATION IN GBFS) ==== -->
<SiteFrame version="1.0" id="EXM02">
   <parkings>
      <Parking version="any" id="station12">
         <Name lang="en">SE Belmont and SE 10 th</Name>
         <Description>Virtual station</Description>
         <gml:Polygon gml:id="polygon_01"> <!-- GBFS: station_area -->
            <gml:name>MultiPolygon</gml:name>
            <gml:interior>
               <gml:LinearRing>
```

```
                <gml:pos srsName="wgs84" srsDimension="2">-122.578067 45.562982</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.661838 45.562741</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.661151 45.504542</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.578926 45.504662</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.578067 45.562982</gml:pos>
                </gml:LinearRing>
             </gml:interior>
          </gml:Polygon>
          <Covered>outdoors</Covered>
          <facilities>
             <SiteFacilitySet version="any" id="station12">
                <Staffing>unmanned</Staffing>
             </SiteFacilitySet>
          </facilities>
          <TotalCapacity>16</TotalCapacity>
          <parkingProperties>
             <ParkingProperties version="any" id="station12>
                <ParkingVehicleTypes>cycle motorScooter car</ParkingVehicleTypes>
                <ParkingVisibility>unmarked</ParkingVisibility>       <!--       implies       GBFS
is_virtual_station -->
             </ParkingProperties>
          </parkingProperties>
          <parkingAreas>
             <VehicleSharingParkingArea version="any" id="station12_01">
                <TotalCapacity>8</TotalCapacity>
                <ParkingProperties id="station12_01" version="any">
                   <vehicleTypes>
                      <SimpleVehicleTypeRef ref="abc123"></SimpleVehicleTypeRef>
                   </vehicleTypes>
                </ParkingProperties>
             </VehicleSharingParkingArea>
             <VehicleSharingParkingArea version="any" id="station12_02">
                <TotalCapacity>8</TotalCapacity>
                <ParkingProperties id="station12_02" version="any">
                   <vehicleTypes>
                      <SimpleVehicleTypeRef ref="def456"></SimpleVehicleTypeRef>
                   </vehicleTypes>
                </ParkingProperties>
             </VehicleSharingParkingArea>
          <VehicleSharingParkingArea version="any" id="station12_03">
                <TotalCapacity>16</TotalCapacity>
                <ParkingProperties id="station12_03" version="any">
                   <vehicleTypes>
                      <SimpleVehicleTypeRef ref="ghi789"></SimpleVehicleTypeRef>
                        </vehicleTypes>
                </ParkingProperties>
             </VehicleSharingParkingArea>
          </parkingAreas>
       </Parking>
    </parkings>
</SiteFrame>
```

A.6.6 system_hours.json – Example

GBFS file: *system_hours.json*.

```
{
       "last_updated": 1609866247,
       "ttl": 86400,
       "version": "2.2",
       "data": {
              "rental_hours": [
                        {
                                "user_types": [ "member" ],
                                "days": ["sat", "sun"],
                                "start_time": "00:00:00",
                                "end_time": "23:59:59"
                        },
                        {
                                "user_types": [ "nonmember" ],
                                "days": ["sat", "sun"],
                                "start_time": "05:00:00",
                                "end_time": "23:59:59"
                        },
                        {
                                "user_types": [ "member", "nonmember" ],
                                "days": ["mon", "tue", "wed", "thu", "fri"],
                                "start_time": "00:00:00",
                                "end_time": "23:59:59"
                        }
                ]
        }
}
```

With NeTEx:

Note that in the example the AVAILABILITY CONDITIONs and DAY TYPEs are declared in line within the relevant entity. Usually the same AVAILABILITY CONDITIONs will apply to many stations so more normal practice would be to declare the AVAILABILITY CONDITIONs separately and the DAY TYPEs in a SERVICE CALENDAR FRAME and reference them.

```
<SiteFrame version="1.0" id="EXM0">
   <Name>Parking site</Name>
   <parkings>
      <Parking version="any" id="pga">
         <Name lang="en">Parking garage A</Name>
         <Description>Physical station</Description>
         <Centroid>
            <Location id="l_01">
              <Longitude>12.345678</Longitude>
              <Latitude>45.678901</Latitude>
            </Location>
         </Centroid>
         <parkingProperties>
            <ParkingProperties version="any" id="pga_01">
              <validityConditions>
                    <AvailabilityCondition id="pga_01" version="any">
                       <dayTypes>
                          <DayType version="any" id="dt_01">
                             <properties>
                                <PropertyOfDay>
                                   <DaysOfWeek>Saturday Sunday</DaysOfWeek>
                                </PropertyOfDay>
                             </properties>
                          </DayType>
                       </dayTypes>
                       <timebands>
                          <Timeband version="any" id="tb_01">
                             <StartTime>00:00:00</StartTime> <!-- GBFS start_time -->
                             <EndTime>23:59:59</EndTime> <!-- GBFS end_time -->
                          </Timeband>
                       </timebands>
                    </AvailabilityCondition>
              </validityConditions>
              <ParkingUserTypes>registered</ParkingUserTypes> <!-- GBFS user_types -->
            </ParkingProperties>
            <ParkingProperties version="any" id="pga_02">
              <validityConditions>
                    <AvailabilityCondition id="pga_02" version="any">
                       <dayTypes>
                          <DayType version="any" id="dt_02">
                             <properties>
                                <PropertyOfDay>
                                   <DaysOfWeek>Saturday Sunday</DaysOfWeek>
                                </PropertyOfDay>
                             </properties>
                          </DayType>
                       </dayTypes>
                       <timebands>
                          <Timeband version="any" id="tb_02">
                             <StartTime>05:00:00</StartTime> <!-- GBFS start_time -->
                             <EndTime>23:59:59</EndTime> <!-- GBFS end_time -->
                          </Timeband>
                       </timebands>
                    </AvailabilityCondition>
              </validityConditions>
              <ParkingUserTypes>other</ParkingUserTypes> <!-- GBFS user_types -->
            </ParkingProperties>
            <ParkingProperties version="any" id="pga_03">
              <validityConditions>
                    <AvailabilityCondition id="pga_03" version="any">
                       <dayTypes>
                          <DayType version="any" id="dt_03">
                             <properties>
                                <PropertyOfDay>
                                   <DaysOfWeek>Monday      Tuesday      Wednesday      Thursday
Friday</DaysOfWeek> <!-- GBFS days -->
                                </PropertyOfDay>
                             </properties>
                          </DayType>
                       </dayTypes>
                       <timebands>
                          <Timeband version="any" id="tb_03">
                             <StartTime>00:00:00</StartTime> <!-- GBFS start_time -->
                             <EndTime>23:59:59</EndTime> <!-- GBFS end_time -->
                          </Timeband>
                       </timebands>
                    </AvailabilityCondition>
              </validityConditions>
              <ParkingUserTypes>allUsers</ParkingUserTypes> <!-- GBFS user_types -->
            </ParkingProperties>
        </parkingProperties>
      </Parking>
   </parkings>
</SiteFrame>
```

A.6.7 system_calendar.json – Example

GBFS file: *system_calendar.json.*

```
{
       "last_updated":1604333830,
       "ttl":86400,
       "version": "2.2",
       "data":{
              "calendars":[
                     {
                            "start_month":4,
                            "start_day":1,
                            "start_year":2020,
                            "end_month":11,
                            "end_day":5,
                              "end_year":2020
                        }
                ]
        }
}
```

With NeTEx:

```
<ServiceCalendarFrame version="1.0" id="scf_01">
        <ServiceCalendar version="any" id="sc_01">
                <FromDate>2021-01-01</FromDate>
                <ToDate>2021-12-31</ToDate>
        </ServiceCalendar>
</ServiceCalendarFrame>
```

A.6.8 system_pricing_plans.json – Example

GBFS file: *system _pricing_plans.json*.

```
{
        "last_updated": 1609866247,
        "ttl": 0,
        "version": "2.2",
        "data": {
                "plans": [
                        {
                                "plan_id": "plan2",
                                "name": "One-Way",
                                "currency": "USD",
                                "price": 2,
                                "is_taxable": false,
                                "description": "Includes 10km, overage fees apply after 10km.",
                                "per_km_pricing": [
                                        {
                                                "start": 10,
                                                "rate": 1,
                                                "interval": 1,
                                                "end": 25
                                        },
                                        {
                                                "start": 25,
                                                "rate": 0.5,
                                                "interval": 1
                                        },
                                        {
                                                "start": 25,
                                                "rate": 3,
                                                "interval": 5
                                        }
                                ]
                        }
                ]
        }
}
```

A.6.8.1 With NeTEx: Simple Price Mapping

This shows just the FARE PRODUCT, GEOGRAPHIC INTERVALs, and GEOGRAPHIC INTERVAL PRICEs. Needed to encode the equivalent data items to the GBFS pricing plan A Full NeTEx declaration would include the TARIFF structure, a SALES OFFER PACKAGE etc.

```
<FareFrame version="1.0" id="plan2">
   <FrameDefaults>
      <DefaultCurrency>EUR</DefaultCurrency> <!-- GBFS currency -->
   </FrameDefaults>
   <geographicalIntervals>
      <GeographicalInterval version="any" id="gi_01">
         <StartGeographicalValue>10</StartGeographicalValue> <!-- GBFS start -->
         <EndGeographicalValue>25</EndGeographicalValue> <!-- GBFS end -->
         <NumberOfUnits>1</NumberOfUnits> <!-- GBFS rate -->
      </GeographicalInterval>
      <GeographicalInterval version="any" id="gi_02">
         <StartGeographicalValue>25</StartGeographicalValue>
         <NumberOfUnits>1</NumberOfUnits>
      </GeographicalInterval>
      <GeographicalInterval version="any" id="gi_03">
         <StartGeographicalValue>25</StartGeographicalValue>
         <NumberOfUnits>5</NumberOfUnits>
      </GeographicalInterval>
   </geographicalIntervals>

   <fareProducts>
      <PreassignedFareProduct version="any" id="plan2">
         <Name>One-Way</Name>
         <Description>Includes 10km, overage fees apply after 10km.</Description>
         <ProductType>singleTrip</ProductType>
      </PreassignedFareProduct>
   </fareProducts>

   <fareTables>
      <FareTable version="any" id="plan2">
         <pricesFor>
              <PreassignedFareProductRef version="any" ref="plan2"/>
         </pricesFor>
         </prices>
           <GeographicalIntervalPrice version="any" id="gi_01">
              <Amount>1.00</Amount>
              <GeographicalIntervalRef version="any" ref="gi_01"/>
           </GeographicalIntervalPrice>
           <GeographicalIntervalPrice version="any" id="gi_02">
              <Amount>0.50</Amount>
              <GeographicalIntervalRef version="any" ref="gi_02"/>
           </GeographicalIntervalPrice>
           <GeographicalIntervalPrice version="any" id="gi_03">
              <Amount>3.00</Amount>
              <GeographicalIntervalRef version="any" ref="gi_03"/>
           </GeographicalIntervalPrice>

           <FareProductPrice version="any" id="plan2_01">
              <Amount>2.00</Amount>
              <PreassignedFareProductRef version="any" ref="plan2"/>
           </FareProductPrice>
         </prices>
      </FareTable>
   </fareTables>
</FareFrame>
```

A.6.8.2 With NeTEx: Extended Product **Mapping**

A full NeTEx declaration of a FARE PRODUCT defines not just the prices.

- A TARIFF structure comprising one or more FARE STRUCTURE ELEMENTs, declares the priced features that make up the product.

- One or more VALIDABLE ELEMENTS indicate which combinations of access rights are available in the FARE PRODUCT.
- A SALES OFFER PACKAGE packages one or more FARE PRODUCTS into an offer that can be purchased, adding distribution information, etc. . The same product might be packaged as deifferent meda (App, Smartcard, paper ticket) and be sold through different channels Etc.

```
<FareFrame version="any" id="plan2">
   <FrameDefaults>
      <DefaultCurrency>EUR</DefaultCurrency> <!-- GBFS currency -->
   </FrameDefaults>
   <tariffs>
      <Tariff version="any" id="plan2">
         <VehicleSharingServiceRef version="any" ref="plan2">
         <geographicalIntervals>
           <GeographicalInterval version="any" id="gi_01">
              <StartGeographicalValue>10</StartGeographicalValue> <!-- GBFS start -->
              <EndGeographicalValue>25</EndGeographicalValue> <!-- GBFS end -->
              <NumberOfUnits>1</NumberOfUnits> <!-- GBFS rate -->
           </GeographicalInterval>
           <GeographicalInterval version="any" id="gi_02">
              <StartGeographicalValue>25</StartGeographicalValue>
              <NumberOfUnits>1</NumberOfUnits>
           </GeographicalInterval>
           <GeographicalInterval version="any" id="gi_03">
              <StartGeographicalValue>25</StartGeographicalValue>
              <NumberOfUnits>5</NumberOfUnits>
           </GeographicalInterval>
         </geographicalIntervals>
         <fareStructureElements>
           <FareStructureElement version="any" id="plan2_access">
              <TypeOfFareStructureElementRef ref="fxc:access"/>
              <geographicalIntervals>
                 <GeographicalIntervalRef version="any" ref="gi_01"/>
                 <GeographicalIntervalRef version="any" ref="gi_02"/>
                 <GeographicalIntervalRef version="any" ref="gi_03"/>
              </geographicalIntervals>
           </FareStructureElement>
         </fareStructureElements>
      </Tariff>
   </tariffs>
   <fareProducts>
      <PreassignedFareProduct version=" any " id="plan2">
         <Name>One-Way</Name>
         <Description>Includes 10km, overage fees apply after 10km.</Description>

           <validableElements>
              <ValidableElement version="any" id="plan2_travel">
                 <Name>Single ride</Name>
                 <fareStructureElements>
                      <FareStructureElementRef version="any" ref="plan2_access"/>
                 </fareStructureElements>
              </ValidableElement>
           </validableElements>

           <accessRightsInProduct>
              <AccessRightInProduct version="any" id="plan2" order="1”>
                 <ValidableElementRef version="any" ref="plan2_travel"/>
              </AccessRightInProduct>
           </accessRightsInProduct>
         <ProductType>singleTrip</ProductType>
      </PreassignedFareProduct>
   </fareProducts>
   <salesOfferPackages>
      <SalesOfferPackage version="any" id="plan2">
         <BrandingRef version="any" ref="EXM01"/>
            <Name>One-Way</Name>
         <distributionAssignments>
            <DistributionAssignment version="any" id="plan2" order="1">
            <DistributionChannelType>atStop</DistributionChannelType>
               <TicketingServiceFacilityList>purchase</TicketingServiceracilityList>
               <PaymentMethods>debitCard creditCard epayDevice</PaymentMethods>
            </DistributionAssignment>
         </distributionAssignments>
         <salesOfferPackageElements>
            <SalesOfferPackageElement version="any" id="plan2" order="1">
                  <TypeOfTravelDocumentRef   ref="etoken"/>
                  <PreassignedFareProductRef version="any" ref="plan2"/>
            </SalesOfferPackageElement>
                  </salesOfferPackageElements>
      </SalesOfferPackage>
   </salesOfferPackages>

   <fareTables>
      <FareTable version="any" id="plan2">
         <pricesFor>
               <PreassignedFareProductRef version="any" ref="plan2"/>
         </pricesFor>
         </prices>
            <GeographicalIntervalPrice version="any" id="gi_01">
               <Amount>1.00</Amount>
               <GeographicalIntervalRef version="any" ref="gi_01"/>
            </GeographicalIntervalPrice>
            <GeographicalIntervalPrice version="any" id="gi_02">
               <Amount>0.50</Amount>
               <GeographicalIntervalRef version="any" ref="gi_02"/>
            </GeographicalIntervalPrice>
            <GeographicalIntervalPrice version="any" id="gi_03">
               <Amount>3.00</Amount>
               <GeographicalIntervalRef version="any" ref="gi_03"/>
            </GeographicalIntervalPrice>

            <FareProductPrice version="any" id="plan2_01">
               <Amount>2.00</Amount>
               <PreassignedFareProductRef version="any" ref="plan2"/>
            </FareProductPrice>
         </prices>
      </FareTable>
   </fareTables>
</FareFrame>
```

A.6.9 geofencing_zones.json – Example

GBFS file: *geofencing_zones.json*.

```
{
       "last_updated":1604198100,
       "ttl":60,
       "version": "2.2",
       "data":{
              "geofencing_zones":{
                     "type":"FeatureCollection",
                     "features":[
                            {
                                   "type":"Feature",
                                   "geometry":{
```

```
                        "type":"MultiPolygon",
                        "coordinates":[
                            [
                                [
                                    [
                                        -122.578067,
                                        45.562982
                                    ],
                                    [
                                        -122.661838,
                                        45.562741
                                    ],
                                    [
                                        -122.661151,
                                        45.504542
                                    ],
                                    [
                                        -122.578926,
                                        45.5046625
                                    ],
                                    [
                                        -122.578067,
                                        45.562982
                                    ]
                                ]
                            ],
                            [
                                [
                                    [
                                        -122.650680,
                                        45.548197
                                    ],
                                    [
                                        -122.650852,
                                        45.534731
                                    ],
                                    [
                                        -122.630939,
                                        45.535212
                                    ],
                                    [
                                        -122.630424,
                                        45.548197
                                    ],
                                    [
                                        -122.650680,
                                        45.548197
                                    ]
                                ]
                            ]
                        ]
                    },
                    “properties":{
                        "name":"NE 24th/NE Knott",
                        "start":1593878400,
                        "end":1593907260,
                        "rules":[
                            {
                                "vehicle_type_id":[
                                    “moped1",
                                    “car1"
                                ],
                                "ride_allowed":false,
                                "ride_through_allowed":true,
                                "maximum_speed_kph":10
                            }
                        ]
                    }
                }
            ]
```

```
                }
        }
}
```

With NeTEx:

```xml
<MobilityServiceFrame version="1.0" id="msf_01">
   <prerequisites>
      <ResourceFrameRef version="1.0" ref="rf_01"/>
   </prerequisites>
   <!-- === GEOFENCING (geofencing_zones.json) === -->
   <mobilityServiceConstraintZones>
      <MobilityServiceConstraintZone version="any" id="mscz_01">
         <validityConditions>
            <AvailabilityCondition version="any" id=" mscz_01">
               <Name>NE 24th/NE Knott</Name> <!-- GBFS: name -->
               <FromDate>2020-01-01T00:00:00Z</FromDate> <!-- GBFS: start -->
               <ToDate>2021-01-01T00:00:00Z</ToDate> <!-- GBFS: end -->
            </AvailabilityCondition>
         </validityConditions>
         <Name>Use of zone</Name>
         <types>
            <TypeOfZoneRef ref="tzr_FeatureCollection"/> <!-- GBFS: type -->
         </types>
         <gml:Polygon gml:id="polygon_01"> <!-- GBFS: geometry -->
            <gml:name>Polygon1</gml:name>
            <gml:interior>
               <gml:LinearRing> <!-- GBFS: coordinates -->
                <gml:pos srsName="wgs84" srsDimension="2">-122.578067 45.562982</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.661838 45.562741</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.661151 45.504542</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.578926 45.5046625</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.578067 45.562982</gml:pos>
               </gml:LinearRing>
            </gml:interior>
            <gml:interior>
               <gml:LinearRing> <!-- GBFS: coordinates -->
                <gml:pos srsName="wgs84" srsDimension="2">-122.650680 45.548197</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.650852 45.534731</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.630939 45.504542</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.630424 45.548197</gml:pos>
                <gml:pos srsName="wgs84" srsDimension="2">-122.650680 45.548197</gml:pos>
               </gml:LinearRing>
            </gml:interior>
         </gml:Polygon>
         <vehicleRestrictions>
            <VehicleTypeZoneRestriction>
               <ZoneUse>forbiddenZone</ZoneUse>     <!-- GBFS ride_allowed -->
            </VehicleTypeZoneRestriction>
            <VehicleTypeZoneRestriction>
               <ZoneUse>allUsesAllowed</ZoneUse>    <!-- GBFS ride_through_allowed -->
            </VehicleTypeZoneRestriction>
            <VehicleTypeZoneRestriction>
               <MaximumSpeed>10</MaximumSpeed>      <!-- GBFS maximum_speed_kph -->
            </VehicleTypeZoneRestriction>
            <VehicleTypeZoneRestriction>
               <SimpleVehicleTypeRef ref="moped1"/> <!-- GBFS vehicle_type_id -->
            </VehicleTypeZoneRestriction>
            <VehicleTypeZoneRestriction>
               <SimpleVehicleTypeRef ref="car1"/> <!-- GBFS vehicle_type_id -->
            </VehicleTypeZoneRestriction>
         </vehicleRestrictions>
      </MobilityServiceConstraintZone>
   </mobilityServiceConstraintZones>
</MobilityServiceFrame>
```

A.7 Examples of mapping – GBFS to SIRI

In this section the GBFS json files that exchange real-time statuses and their corresponding SIRI mappings are presented.

A.7.1 station_status.json – Example

The following code fragments show an example of a station status query in GBFS and in SIRI formats.

GBFS: *station_status.json*

```
{
"last_updated": 1434054678,
"ttl": 0,
"version": "v2.1-RC",
"data": {
        "stations": [
        {
                "station_id": "station1",
                "is_installed": true,
                "is_renting": true,
                "is_returning": true,
                "last_reported": 1434054678,

                "num_docks_available": 3,

        "num_bikes_available": 1,
                "vehicle_types_available": [{
                "vehicle_type_id": "abc123",
                "count": 1
                }, {
                "vehicle_type_id": "def456",
                "count": 0
                }]

                "vehicle_docks_available": [{
                "vehicle_type_ids": ["standardbike"],
                "count": 2
                },{
                "vehicle_type_ids": ["escoot3"],
                "count": 1
                }]
        },
```

As SIRI (Facility Monitoring)

SIRI Header (note that this type of header is applicable to all other SIRI example, and can be much more detailed if necessary)

```
<FacilityMonitoringDelivery namespaces, etc.>

        <ResponseTimestamp>2001-12-17T09:30:47Z</ResponseTimestamp>
        <RequestMessageRef>134567-</RequestMessageRef>
        <Status>true</Status>
        <ValidUntil>2001-12-17T09:30:47Z</ValidUntil>
        <ShortestPossibleCycle>P1Y2M2DT10H30M</ShortestPossibleCycle>
```

SIRI Payload

```
<FacilityCondition>
        <FacilityRef>MYNS:ParkingArea:station2</FacilityRef>

        <FacilityStatus>
                <Status>available</Status>
        </FacilityStatus>

        <MonitoredCounting> <!--this bloc isn't mandatory since their is also a detail per type
of bay-->
                <CountingType>availabilityCount</CountingType>
                <CountedFeatureUnit>bays</CountedFeatureUnit>
                <Count>3</Count>
        </MonitoredCounting>

        <MonitoredCounting>
                <CountingType>availabilityCount</CountingType>
                <CountedFeatureUnit>vehicles</CountedFeatureUnit>
                <TypeOfCountedFeature>
                        <TypeOfValueCode>abc123</TypeOfValueCode>
                        <NameOfClass>bike</NameOfClass> <!--not mandatory, but helps-->
                </TypeOfCountedFeature>
                <Count>1</Count>
        </MonitoredCounting>

        <MonitoredCounting>
                <CountingType>availabilityCount</CountingType>
                <CountedFeatureUnit>vehicles</CountedFeatureUnit>
                <TypeOfCountedFeature>
                        <TypeOfValueCode>def456</TypeOfValueCode>
                        <NameOfClass>bike</NameOfClass>
                </TypeOfCountedFeature>
                <Count>0</Count>
        </MonitoredCounting>

        <MonitoredCounting>
                <CountingType>availabilityCount</CountingType>
                <CountedFeatureUnit>bays</CountedFeatureUnit>
                <TypeOfCountedFeature>
```

```xml
				<TypeOfValueCode>escoot3</TypeOfValueCode>
				<NameOfClass>escooter</NameOfClass>
			</TypeOfCountedFeature>
			<Count>1</Count>
		</MonitoredCounting>

		<MonitoredCounting>
			<CountingType>availabilityCount</CountingType>
			<CountedFeatureUnit>bays</CountedFeatureUnit>
			<TypeOfCountedFeature>
				<TypeOfValueCode>standardbike</TypeOfValueCode>
				<NameOfClass>bike</NameOfClass>
			</TypeOfCountedFeature>
			<Count>0</Count>
		</MonitoredCounting>

		<ValidityPeriod>
			<StartTime>2001-12-17T09:30:47Z</StartTime> <!--for "last_updated"-->
		</ValidityPeriod>

</FacilityCondition>
```

A.7.2 *free_bike_status.json* – Example

The following code fragments show an example of a free bike status query in GBFS and in SIRI formats.

```json
{
	"last_updated": 1434054678,
	"ttl": 0,
	"version": "v2.1-RC",
	"data": {
		"bikes": [
			{
				"bike_id": "ghi789",
				"last_reported": 1434054678,
				"lat": 12.3499,
				"lon": 56.7899,
				"is_reserved": false,
				"is_disabled": false,
				"vehicle_type_id": "standardbike"

			}, {
				"bike_id": "jkl012",
				"last_reported": 1434054687,
				"lat": 12.3499,
				"lon": 56.7899,
				"is_reserved": false,
				"is_disabled": false,
				"vehicle_type_id": "escoot3",
				"current_range_meters": 6543
			}
		]
	}
}
```

With SIRI (Facility Monitoring), Vehicles at station are a specific case of the free bike status in GBFS:

```
                        {
                                "bike_id": "ghi789",
                                "last_reported": 1434054678,

                                .
                                "vehicle_type_id": "standardbike"
                                "station_id": "ghi789",
                        },
```

SIRI Payload

For SIRI, information about vehicles at a station is included with the station information.

```
<siri:FacilityCondition>
                <siri:FacilityRef>SOMEOPERATOR:Vehicle:ghi789</siri:FacilityRef>
        <!--If the vehicle is not described in the static data, the Description element can be used to describe it-->

                <siri:FacilityStatus>
                        <siri:Status>available</siri:Status>
                </siri:FacilityStatus>

                <siri:MonitoredCounting>
                        <siri:CountingType>currentStateCount</siri:CountingType>    <!--can    be
availabilityCount, inUseCount, outOfOrderCount, reservedCount ...-->
                        <siri:CountedFeatureUnit>meters</siri:CountedFeatureUnit>
                        <siri:TypeOfCountedFeature>
                                <siri:TypeOfValueCode>CurrentRange</siri:TypeOfValueCode>
<!--note: profile or user defined code, could also be CurrentChargeLevel, etc.-->
                                <siri:NameOfClass>Vehicle</siri:NameOfClass>
                        </siri:TypeOfCountedFeature>
                        <siri:Count>6543</siri:Count>
                </siri:MonitoredCounting>

                <siri:FacilityUpdatedPosition>
                        <siri:Longitude>56.7899</siri:Longitude>
                        <siri:Latitude>12.3499</siri:Latitude>
                </siri:FacilityUpdatedPosition>

        </siri:FacilityCondition>
```

Vehicles at station are a specific case of the free bike status in GBFS

```
        {
                "bike_id": "ghi789",
                "last_reported": 1434054678,


                "vehicle_type_id": "standardbike"
                "station_id": "ghi789",
        },
```

For SIRI, information about vehicles at a station is enclosed within the station information.

```
        <FacilityCondition>
                <FacilityRef>SOMEWHERE:ParkingArea:station2</FacilityRef>

                <FacilityStatus>
                        <Status>available</Status>
                </FacilityStatus>

                <MonitoredCounting> <!--One count per type of vehicle (if type is provided)-->
                        <CountingType>availabilityCount</CountingType>
                        <CountedFeatureUnit>vehicles</CountedFeatureUnit>
                        <TypeOfCountedFeature>
                                <TypeOfValueCode>standardbike</TypeOfValueCode>
                                <NameOfClass>bike</NameOfClass>
                        </TypeOfCountedFeature>
                        <Count>1</Count>
                        <CountedItemsIdList>
                                <ItemId>ghi789</ItemId>
                        </CountedItemsIdList>
                </MonitoredCounting>

                <MonitoredCounting> <!--One count per type of vehicle -->
                        <CountingType>availabilityCount</CountingType>
                        <CountedFeatureUnit>vehicles</CountedFeatureUnit>
                        <TypeOfCountedFeature>
                                <TypeOfValueCode>escoot3</TypeOfValueCode>
                                <NameOfClass>escoot</NameOfClass>
                        </TypeOfCountedFeature>
                        <Count>2</Count>
                        <CountedItemsIdList>
                                <ItemId>vwx234</ItemId>
                                <ItemId>vwx235</ItemId>
                        </CountedItemsIdList>
                </MonitoredCounting>
        </FacilityCondition>
</FacilityMonitoringDelivery>
```

A.7.3 *system_alerts*.json – Example

The following code fragments show an example of system_alerts query in GBFS and in SIRI formats.

GBFS file: *system _alerts.json.*

```
{ "last_updated":1604198100,l":60,
        "ttl
        "data":{
                "alerts":[
                        {
                                "alert_id":"21",
                                "type":"station_closure",
                                "station_ids":[
                                        "123",
                                        "456",
                                        "789"
                                ],
                                {
                                                "end":"1604674800"
                                        }
                                ],
                                "summary":"Disruption of Service",
                                "description":"The  three  stations  on  Broadway  will  be  out  of
service
                                 from 12:00am Nov 3 to 3:00pm Nov 6th to accommodate road work",
                                "last_updated":1604519393
                        }
                ]
```

With SIRI (Situation Exchange)

```
<PtSituationElement>
   <CreationTime>2001-12-17T09:30:47Z</CreationTime>
   <SituationNumber>GBFS:PtSituationElement:21</SituationNumber>
   <Source>
        <SourceType>directReport</SourceType> <!--mandatory-->
   </Source>
   <VersionedAtTime>2020-01-16T13:00:00Z</VersionedAtTime>
   <Progress>open</Progress>

   <ValidityPeriod>
        <StartTime>2020-01-16T13:00:00Z</StartTime>
        <EndTime>2020-01-16T14:00:00Z</EndTime>
   </ValidityPeriod>

   <MiscellaneousReason>accident</MiscellaneousReason>

   <Summary>Disruption of Service</Summary>
   <Description xml:lang="en-us">The three stations on Broadway will be out of service
   </Description>

   <InfoLinks>
        <InfoLink><Uri>https://example.com/more-info</Uri>
        </InfoLink>
   </InfoLinks>

   <Affects>
        <!--Can be refined by Operator, Network ... -->
        <Places>
                <AffectedPlace>
                        <PlaceRef>GBFS:station:123</PlaceRef>
                </AffectedPlace>
                <AffectedPlace>
                        <PlaceRef>GBFS:station:456</PlaceRef>
                </AffectedPlace>
                <AffectedPlace>
                        <PlaceRef>GBFS:station:789</PlaceRef>
                </AffectedPlace>
                <AffectedPlace>
                        <PlaceRef>GBFS:Region:region_id</PlaceRef>    <!--can    also    be    a
TopographicPlace Id: useful for the GBFS region_ids-->
                </AffectedPlace>
        </Places>
   </Affects>

   <Consequences>
        <!--can have a specific Affects if different from the Situation Affects, can also host
multiple consequences-->
        <Consequence>
                <Condition>noService</Condition>   <!—can   also   be   "altered",   "diverted",
"normalService", "stopMoved", "disruption", "undefinedServiceInformation" -->
        </Consequence>
   </Consequences>
```

```xml
</PtSituationElement>
```